Hybrid techniques for speech coding

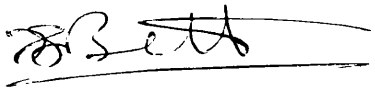
Burnett, I. S.

Award date:
1992

Awarding institution:
University of Bath

Hybrid Techniques for Speech Coding

Submitted by I. S. Burnett
for the degree of PhD
of the University of Bath
1992.



I. S. Burnett

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Summary

Speech coding techniques generally operate either in the Time or Frequency domain. This thesis considers new, hybrid Linear Predictive coding (LPC) architectures which operate in both domains. Such schemes are shown to be capable of producing coded speech of both higher perceptual quality and at lower bit rates than conventional single-domain coders.

The basis of the work is an adapted Analysis-by-Synthesis (A-by-S) architecture which searches gaussian codebooks in the Discrete Frequency domain. Frequency domain searching offers computational advantages but Full Frequency domain codebooks are too large for practical coder implementations. A new Overlapped Frequency Domain codebook is thus described which requires one fourtieth of the memory space of a standard Full Frequency domain codebook. A further consequence of the Frequency domain A-by-S architecture is that a 'pseudo-ideal' excitation sequence can be derived. Analysis of this 'pseudo-ideal' excitation leads to new, improved coding architectures.

An advantage of Frequency Domain searching is that perceptual weighting becomes a simple vector multiplication. This allows the incorporation of an improved perceptual measure, the Bark Spectral Distortion (BSD), into the A-by-S architecture. The BSD is a perceptual measure which models the psycho-acoustic properties of the human ear and is shown to improve the perceptual quality of coded speech. Further quality improvements are generated by using an improved resolution BSD which simulates the ear more closely.

Analysis of the excitation shows that the pitch of voiced speech is not well represented by the standard A-by-S architecture. To code voiced speech at

lower bit-rates a new scheme is proposed, which uses a series of 'pitch-prototypes' to produce the required pitch periodicity. The prototype waveform (PW) coder interpolates the 'pitch-prototypes' in the Frequency domain and two prototype quantisation schemes are presented. In combination with a standard CELP scheme for unvoiced frames, the PW/CELP coder is shown to produce good, coded speech quality at sub-3.2kbit/s.

Table of Contents

Glossary of Acronyms

The following acronyms are used throughout this thesis:

A-by-S	Analysis-by-Synthesis
AV.SNR	Average Signal-to-Noise Ratio
BSD	Bark Spectral Distortion
CD	Cepstral Distance
CELP	Code Excited Linear Prediction
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
LPC	Linear Predictive Coding
LSF	Line Spectral Frequency
LTP	Long Term ('Pitch') Predictor
MBE	Multi-Band Excitation
MPE	Multi-pulse Excitation
MSE	Mean Squared Error
PW	Prototype Waveform
PW/CELP	Mixed Prototype Waveform/CELP coder
RELP	Residual Excited Linear Prediction
RPE	Regular Pulse Excitation
SBC	Sub-Band Coding
SEGSNR	Segmental Signal-to-Noise Ratio

Chapter 1: Introduction

Speech is man's primary form of communication and it's transmission and storage, by analogue techniques has been fundamental in the technological growth of this century. Speech coding, is the enabling technology behind the move from the traditional anologue media to digital techniques. While at a higher bandwidth, this is perhaps more readily seen in the recording industry where digital formats (e.g. CD, DAT and DCC) are fast supplanting the older analogue media. For speech, the transfer to digital formats brings the advantages of efficient storage and transmission, higher noise immunity and enhanced security. There is an ever-increasing number of new applications requiring these performance advantages, and with each come new requirements and challenges.

1.1 Digital Speech Coding Applications

Modern speech coding can be traced back to the 1930s when the first experiments with Pulse Code Modulation (PCM) were performed. However, current speech coding techniques have been primarily developed over the last 15 years in line with improvements in processor technology.

Much of the early impetus for speech coding came from defence and government sources, who saw speech coding as the key to high security communications. The digital coding of speech allowed the use of digital encryption rather than the, then, primitive analogue scrambling techniques. The coders developed for such applications were very low bit-rate (e.g. the 2.4kbit/s LPC-10 [1] technique), and, while offering high

intelligibility, had poor overall quality, lacking naturalness and speaker recognition. Such problems are tolerable in military circumstances, where listeners are well-trained radio operators, but would not be acceptable to the general public on a telephone network.

Digital telephone networks are at the opposite extreme to the low bit-rate military applications and require very-high perceived speech quality. In such schemes, it is often necessary to cascade coders many times such that an individual coding operation must be near transparent to a listener. There are three CCITT standards for network speech coders operating at 64kbit/s (PCM in 1972), 32kbit/s (ADPCM in 1984) [2], and currently 16kbit/s (LD-CELP) [3]. All these standards, can not only transmit 'Toll-quality' speech, but also non-speech, voice band data signals such as modem tones.

Between the extremes of network and military coders, lie the low/medium rate coders that are the subject of this thesis. These operate between 4.8kbit/s and 16kbit/s and are currently used for such purposes as cellular telephony, military communications and voice-mail. For some of these applications, near-network quality is required, but in others speech quality is compromised for bit-rate or implementation complexity.

In the following sections, current speech coding techniques used for low/medium rate coding and the resulting world-wide coding standards are considered.

1.2 Low/Medium Rate Speech Coding Techniques

The 4.8-16kbit/s speech coding techniques can be divided into two basic categories: those operating in the frequency (or transform) domain and those in the time domain. A simple 'tree' of current coding schemes is shown in Figure 1.1. All the coders exploit knowledge of the nature of

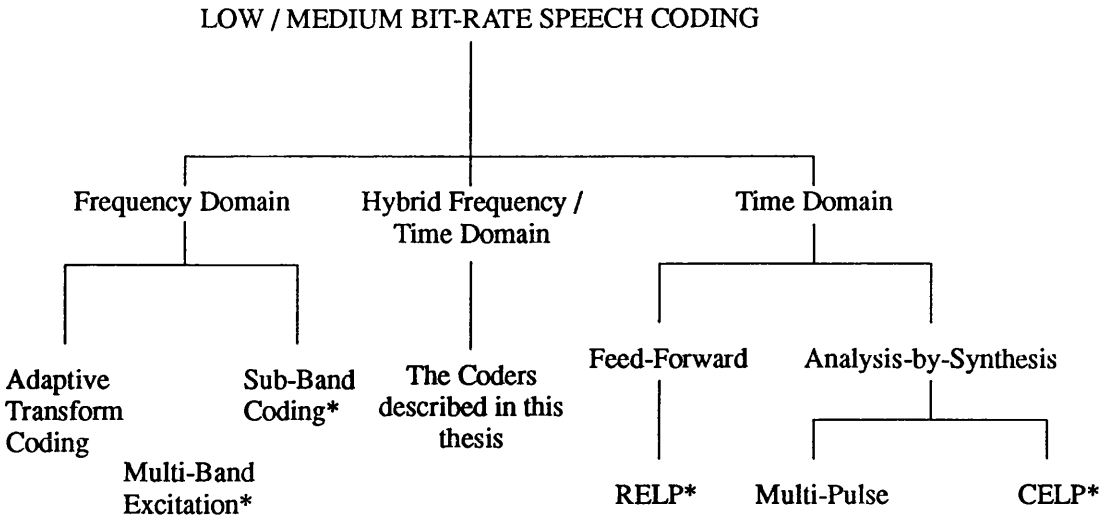


Figure 1.1: A summary of current low/medium bit-rate (<16kbit/s) speech coding techniques. * indicates that a current standard specifies the technique.

speech to remove predictable components and, hence, reduce transmission bit-rates. Each coder type is now briefly considered:

1.2.1 Frequency Domain Coders

Transform coders [4] operate by transforming short time sections of the input speech. Adaptive bit allocation algorithms are then used to code the resulting spectral coefficients. Such coders can produce near-toll quality at 16kbit/s, but deteriorate rapidly at lower rates.

Sub-Band coders [5] are related to transform coders but filter the input speech into a number of bands. The band outputs are then decimated before quantisation, such that the key to high quality sub-band coding is dynamic bit-allocation between filter outputs. Sub-band coders have similar transmission rates to transform coders. An interesting communications-quality variant on sub-band coding is in commercial production and operates at 8kbit/s [6].

Multi-Band Excitation (MBE) coders [7] were introduced recently, and differ from previous frequency domain approaches in the use of the speech pitch. The spectrum is quantised in bands, defined by the pitch harmonics and such schemes can produce good communications quality speech at 6.4kbit/s.

1.2.2 Time Domain Coders

Time domain coders can be further divided into 'feed forward' structures and Analysis-by Synthesis Schemes. A typical example of the former is Residual Excited Linear Predictive coding (RELP) [8]. In such a scheme the predictable parts of the speech are removed and the residual waveform is then directly quantised. Such schemes achieve bit rates of ~ 9.6kbit/s for good communications and 13kbit/s for Toll-quality speech.

Analysis-by-Synthesis (A-by-S) schemes [9] contrast with the feed-forward approach of RELP and synthesise speech from a selection of excitation and predictor parameters. The synthesised speech is then compared with the input speech according to some perceptually meaningful criterion and the optimum set of excitation and predictor parameters are transmitted. Typically, the excitation is selected as a gaussian sequence (as in Code Excited Linear Prediction, CELP) and such schemes can produce near Toll-quality speech at 4.8kbit/s.

1.3 Current speech coding standards

In Figure 1.1 the coder structures denoted * are currently the subject of a world-wide standard for speech coding. From the diagram it can be seen that there is currently little consensus on the correct coding scheme and the standards will be briefly considered in descending bit-rate order:

The first digital cellular scheme to be operational is the GSM Pan-European network - at the time of writing Vodafone plc. have a small system operational in the U.K. with full coverage promised for December 1992. This digital network currently uses the GSM full-rate coder [10], which implements a RELP scheme (including a pitch predictor) and operates at 13kbit/s. With error correction this translates to a gross bit-rate per voice channel of 21kbit/s. Already GSM are performing evaluation tests on a half rate codec (gross bit rate 11.4kbit/s and a coder rate of approximately 6.5kbit/s), which is likely to be a CELP A-by-S architecture. This would compare with the American cellular scheme's full-rate CELP coder operating at 8kbit/s [11].

Operating at a considerably lower rate, is the INMARSAT IMBE (Improved MBE) coder [12], which is designed for digital voice coding over low bit-rate satellite channels. This speech coder operates at 6.4 kbit/s and produces good communications-quality speech.

Finally the US Federal Standard 1016 speech coder [13] is a fully optimised 4.8kbit/s CELP coder. It produces near-Toll quality speech at this bit-rate and is currently under consideration for telephony applications. This coder was originally intended as a replacement for the LPC10-type synthetic quality coders.

The last two standards have only recently been fully completed and thus some work in this thesis pre-dates their full publication.

1.4 Speech Coding Challenges

From this brief discussion, it is clear that there are a variety of current speech coding techniques, however most of the coders are based around A-by-S architectures. These are the basis for the work in this thesis and there are three areas of current A-by-S research:

- Improvements in the perceptual quality of coded speech.
- Reductions in transmission bit-rate.
- Reduction in the coding delay generated by such coders.

This thesis is concerned with the first of these two areas, but delay is a serious problem for many of the current coder applications. Many network standards require maximum coding delays of less than 5ms (e.g. CCITT 16kbit/s), precluding the use of the block algorithms used for A-by-S schemes. Thus for low delay applications new backward adaptive CELP architectures are being researched; these sacrifice bit-rate reductions for substantially reduced coding delays.

In this thesis, the first area considered is improvements in the perceptual quality of CELP coded speech. The key to such improvements is the adoption of a hybrid CELP architecture which mixes both time and discrete frequency domain techniques. The use of the DFT offers computational advantages, but also creates a number of design problems, which are considered in Chapter 3. However, a substantial advantage of the DFT domain is that improved perceptual measures can be employed. These measures simulate the psycho-acoustic / perceptual behaviour of the ear and their incorporation into the standard A-by-S architectures is considered.

The hybrid CELP architecture also reveals a number of limitations of the standard A-by-S architecture and from these results a new low bit-rate coding architecture is conceived. This 'prototype' coder exploits the pitch periodicity of speech, and is capable of substantially reducing transmission bit-rate.

1.5 A Note on Organisation

This thesis is divided into five key theoretical chapters. Chapter 2 considers the human vocal and auditory systems, while Chapter 3 concentrates on the background signal processing for time-domain A-by-S speech coding. From this development a standard Time-Domain CELP architecture is derived, providing a basis for the work of Chapters 4, 5 and 6. Chapter 4 discusses DFT domain CELP architectures [14] and derives a new DFT domain codebook technique. The new DFT domain CELP coder is then adapted in Chapter 5 to use a new perceptual measure, the Bark Spectral Distortion (BSD). Finally, Chapter 6 investigates a new Prototype based coder and derives a sub-3kbit/s coder based on the technique.

1.6 References

[1] T. E. Tremain, "The government standard linear predictive coding algorithm: LPC-10," *Speech Technology*, Vol. 1, No. 2, pp. 40-49, April 1982.

[2] CCITT Study Group XVIII, "32Kb/s adaptive differential pulse code modulation (ADPCM), ," *Working Party 8, Draft revision of recommendation G. 721, Temp. Document D.723/XVIII.*

[3] J. H. Chen, R. V. Cox, Y. C. Lin, N. S. Jayant and M. J. Melchener, "A Low-Delay CELP Coder for the CCITT 16kb/s Speech Coding Standard," *IEEE J. on Sel. Areas in Comms.*, Vol. 10, No. 5, pp. 830-849, June 1992.

[4] R. Zelinski and P. Noll, "Adaptive Transform Coding of Speech Signals," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. 25, No. 4, pp. 299-309, Aug. 1977.

[5] J. M. Tribolet and R. E. Crochiere, "Frequency Domain Coding of Speech," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. 27, No. 5, pp. 512-530, Oct. 1979.

[6] N. G. Kingsbury, "Robust 8000 bit/s sub-band speech coder," *IEE Proceedings*, Vol. 134, Pt. F, No. 4, pp. 352-366, July 1987.

[7] M. Brandstein, J. Hardwick and J. Lim, "The Multi-Band Excitation Speech Coder," *in Advances in Speech Coding, ed. Atal, Cuperman and Gersho*, Kluwer Academic Publishers, pp. 215-224, 1991.

[8] D. O'Shaughnessy, "Speech Communication: Human and Machine," *Addison-Wesley* , pp. 365-370, 1987.

[9] M. R. Schroeder and B. S. Atal, "Code-Excited Linear Prediction (CELP): High-Quality Speech at Very Low Bit Rates," *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proc.*, pp. 937-940, 1985.

[10] European Telecommunications Standards Institute Technical Committee, "Recommendation 06.10: GSM Full-Rate Speech Transcoding," Version 3.2.0, Jan. 1990.

[11] I. A. Gerson and M. A. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8kbps," *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proc.*, pp. 461-464, April 1990.

[12] INMARSAT Council Meeting 1990, "INMARSAT-M Voice Coding Algorithm," , published as SDM\MMOD1\APPENDIX\ISSUE 3.0, August 1991.

[13] U.S. National Communications System, Washington, D.C. , "Proposed Federal Standard 1016, Second Draft," Nov. 1989.

[14] I. S. Burnett and R. J. Holbeche, "The Application of the DFT to CELP Architectures," Proc. *IEEE Workshop on Speech Coding for Telecommunications: Digital Voice for the Nineties*, pp. 83-84, Whistler, B.C., Canada, Sept. 1991.

Chapter 2: The Human Vocal and Auditory Systems.

This chapter describes the roles of the human vocal and auditory systems, since these originate and receive speech, their characteristics are an important consideration in the design of speech coders.

2.1 The Human Vocal System

The basic features of the human vocal system [1][2][3] are shown in Figures 2.1 and 2.2. The human vocal system is unique in being able to produce meaningful speech sounds; these are generated by varying the positions of the vocal tract articulators; the vocal chords, tongue, lips, teeth, velum and jaw. Sounds may be broadly classed into two sets: (a) vowels, which result from unrestricted airflow through the vocal tract and (b) consonants, which are generated by airflow restrictions at one of various points in the vocal tract.

The 'energy source' for speech is the exhalation of air from the lungs. Sounds formed during inhalation are rare and the 1:10 inhalation/exhalation ratio during normal breathing thus enhances speech production. The rate of exhalation from the lungs controls the amplitude of sounds and the shape is defined by the vocal tract obstructions; when no obstruction is present normal breathing occurs.

Most of the vocal tract obstructions, used in speech production, are situated in the larynx where the vocal chords (or folds) can partially or completely obstruct the vocal tract. The folds are structures of muscle, tendon and mucous membrane which can be varied in length, thickness and position by various muscular contractions.

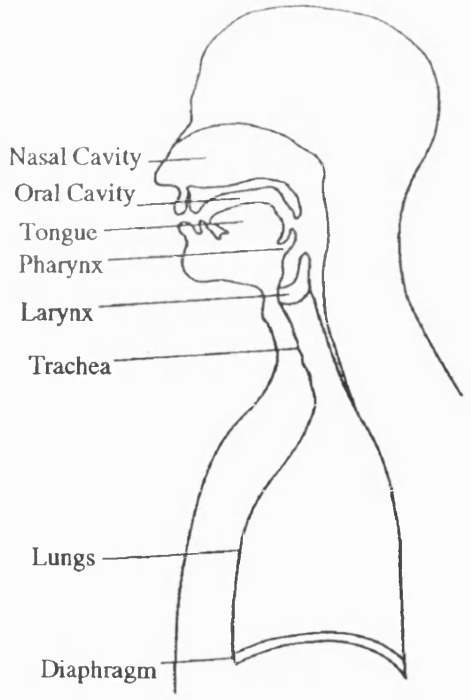


Figure 2.1: The organs used in speech production.

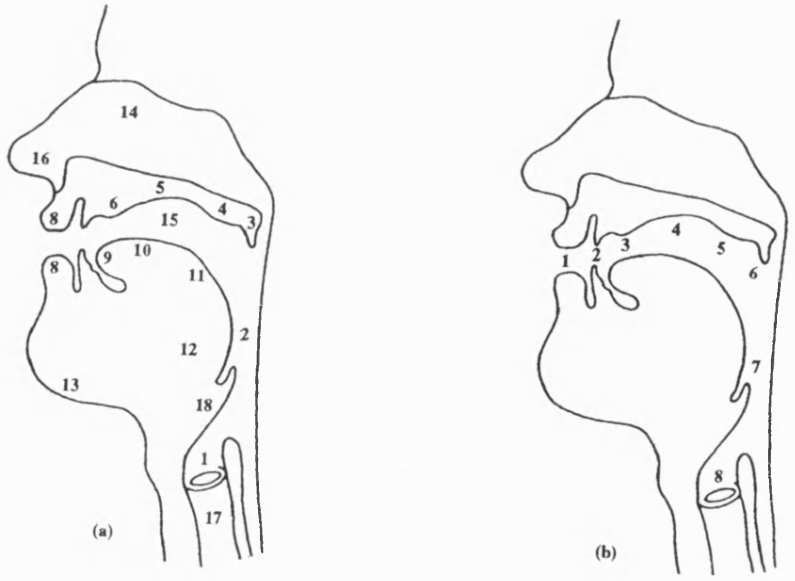


Figure 2.2: Cross-sectional views of (a)vocal tract and (b)places of articulation. Key: (a): 1.vocal fold, 2. pharynx, 3. velum, 4. soft palate, 5. hard palate, 6. alveolar ridge, 7. teeth, 8. lips, 9. tongue tip, 10. blade, 11. dorsum, 12. root, 13. jaw, 14. nasal cavity, 15. oral cavity, 16. nostrils, 17. trachea, 18. epiglottis. (b): 1.labial, 2. dental, 3. alveolar, 4. palatal, 5. velar, 6. uvular, 7. pharyngeal, 8. glottal.

2.1.1 Sound Production.

There are three basic classes of sound produced by the vocal tract:

a) 'Whispers' are produced by almost closing the vocal folds completely so that turbulent noise is generated at the epiglottis. A second set of sounds, known as fricatives ('Shhh', 'Fhhh'), are generated in a similar way, but at a point further up the vocal tract. Fricatives are produced by placing the tongue against the roof of the mouth or the lips against the teeth.

b) 'Stops', or 'Plosives' (/P/, /T/) are caused by airflow interruptions. In this case, the vocal tract is completely closed at one of various points, such as the glottis (at the vocal folds), the tongue (against the palate e.g. /T/), and the lips (e.g. /P/).

c) The final class of sounds are known as 'Voiced' or 'Sonorant' sounds (/i,m/), which are those generated by motion of the vocal folds. These are the most important sounds in speech and are produced by periodic interruptions in the vocal tract airflow. The periodicity is produced by opening and closing of the vocal folds and generates the fundamental frequency or pitch of the speech. The average period of oscillation varies as the size of the vocal folds, which in turn is variant with age and sex of the speaker. Male speakers have long vocal folds and corresponding low fundamental frequencies, while children have short folds generating high fundamental frequencies. Neither the fundamental period, nor the vocal tract shape are constant, meaning that voiced speech is not truly periodic. It can, however, be considered as quasi-periodic over short time intervals

2.1.2 Vocal Tract

The vocal and nasal tracts can be regarded as tubes of non-uniform cross sectional area. The speech sounds are generated by airflow down these tubes, and within the tubes there are a variety of resonance effects. These are similar to those found in organ pipes and other wind instruments. The resonant frequencies of the vocal tract are known as formant frequencies or, simply, formants. These are defined by the shape and size of the vocal tract such that different sounds are characterised by different formants. Thus the spectral properties of the speech signal are controlled by the time-varying structure of the vocal tract.

The vocal tract structures which allow the production of different, discriminatable, sounds are known as the 'Articulators'.

2.1.3 Vocal Tract Articulators

The most important Articulators are the tongue and lips but the velum and larynx also have important roles for some sounds. The larynx controls airflow through the glottis and can also be raised or lowered to: (a) effect formant frequencies, (b) lengthen the vocal fold vibrations, and (c) facilitate movement of the upper articulators.

The most visible of the articulators are the lips. These effect vocal tract closure by producing narrow slits, rounding and spreading. The teeth can also be used in conjunction with the lips , for example in the dental obstruent sound /f/. The teeth are also used to produce sounds such as /θ/, which are generated by actions of the most important articulator, the tongue, against the teeth.

The tongue forms most of the lower wall of the upper section of the vocal tract. It is extremely agile and can exert considerably more influence on sound formation than the structures of the upper wall. To produce

different sounds the tongue is positioned and its shape altered. For these movements the tongue contains four distinct components and a sophisticated muscle structure. These control mechanisms allow the agile tip of the tongue to contact the palate up to nine times per second in tonguing type movements. The muscles also allow the tongue to be repositioned in times of less than 50ms, making the tongue a sophisticated tool for the production of vocal tract constrictions, at a number of positions.

All of the vocal tract articulators can be controlled precisely by the brain. To facilitate the level of control required each articulator has a high neuron to muscle fibre ratio. This allows very precise control of movements which are normally less than 1cm and can be at speeds of up to 30cm/s.

In this brief summary of the role of the human vocal system only the fundamentals have been covered. Further details can be found in the references [1][2]. A myriad of articulatory and phonetic analysis has been performed on the methods of speech production and the disorders thereof. These are beyond the scope of this thesis and the reader is referred to [1][4] for further details. In terms of speech coding, however, the most important conclusion is that speech is a highly complex signal. The generating processes are many and varied. Any coding or synthesis scheme will necessarily make an approximation to the generating processes by modelling the role of the fundamental features, while discarding less important sections. The choice of the model and the importance of the various speech characteristics, are the fundamental decisions required for the design of successful coding and synthesis schemes.

2.2 The Human Auditory System

In the past. communications engineers have often concentrated speech coding efforts on modelling the speech production process. It is, however, becoming increasingly apparent that the human auditory processes [1][5][6][7] should have a position of equal importance in the design of synthesis and coding schemes. The auditory processing consists of both physiological and psychological effects, both of which influence design decisions made in this thesis. Hence, in the following summary of the ear both the psychology and physiology of hearing is considered.

Physiology and Anatomy of the ear

The human ear is, not surprisingly, particularly sensitive to the frequency range of human speech that contains most information (~200-5600Hz). Perception of these sounds is highly detailed and the ear automatically compensates for the fall off in voiced speech energy beyond 400Hz. The ear is divided into three fundamental sections; the outer, middle and inner ear. The outer ear directs sounds towards the eardrum, from where the middle ear transforms sound pressure into mechanical movement. The inner ear, converts these movements into firings of the auditory neurons which, in turn, send electrical signals to the brain.

In the following sections the basic physiology and anatomy of the ear is considered. The diagrams of Figures 2.3 and 2.4 show the basic important structures of the ear.

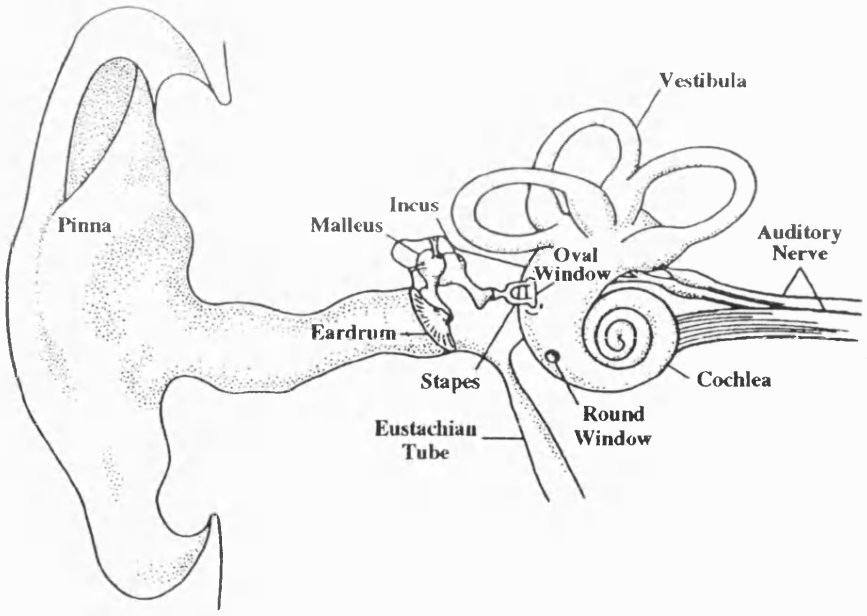


Figure 2.3: The important structures of the human ear.

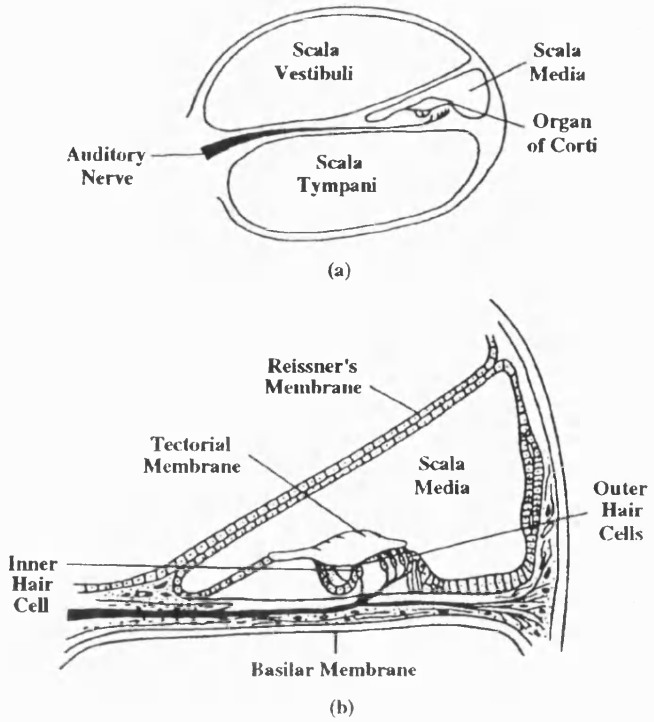


Figure 2.4: The structure of the cochlea: (a) a cross-section of the cochlea and (b) the structures surrounding the basilar membrane and Organ of Corti.

2.2.1 The Outer Ear

The outer ear is substantially visible. This part is known as the pinna (see Figure 2.3) and serves to funnel sounds towards the tympanic membrane (or eardrum). The pinna's shape also allows the ear to be more sensitive to sounds emanating from sources in front of the listener and fulfils the further, important role of protecting the sensitive middle and inner ear from foreign objects.

The air filled cavity formed by the outer ear acts as a 1/4 wavelength resonator with a first resonance, for adults, of ~3kHz. This amplifies the spectrum between 3 and 5kHz by up to 15dB, which significantly improves the perception of high frequency sounds.

2.2.2 The Middle Ear

The resonances generated in the outer ear impinge on the tympanic membrane which is at the beginning of the small air filled cavity known as the middle ear. This contains three tiny bones which transmit the tympanic membrane vibrations to the oval window membrane of the inner ear.

The middle ear performs three major functions:

1. The small bones (ossicles) provide an impedance match between the fluid filled inner ear and the air filled cavity of the outer ear. Without such a matching device only ~0.1% of the energy impinging on the tympanic membrane would reach the inner ear.
2. A further amplification is provided by the 'lever' action of the ossicles, combined with the difference in surface area between the large eardrum and small oval window membrane. The total increase in sound pressure generated by both these devices is approximately 30dB.

3. Protection against dangerously intense sounds that could damage the sensitive inner ear. This is provided by the stapes which alters from a pumping action to rotation at high sound levels. The oscillations of the inner ear do not, therefore, increase proportionally with the impinging sound level. The protection system is also activated (as a reflex action) when speaking. This prevents overloading and damage from the intense sound levels generated by speech at the speakers own ear.

Spectrally, the middle ear acts as a lowpass filter with a roll off of ~15dB / Octave above 1kHz.

2.2.3 The Inner Ear

The cochlea is the central part of the inner ear and is a fluid filled tube which transforms the mechanical vibrations at its 'oval window' into electrical signals sent from the auditory neurons to the brain. As can be seen in Figure 2.3, the cochlea is a snail-like spiral which is divided internally into three liquid filled tubes by two membranes. The three tubes are known as the Scala Vestibuli, the Scala Media, and the Scala Tympani while the dividing membranes are Reissner's membrane and the Basilar membrane.

The stapes of the middle ear transmit vibrations to the Scala Vestibuli via the oval window membrane. The cochlea has solid bone wall and is filled with an incompressible liquid. Thus, the vibrations of the oval window membrane cause motion in the flexible membranes of the cochlea. Two small holes, one at the apex of the cochlea, and the other at the basal end (the round window) serve to relieve pressure in the scalae.

The cochlea has a tapering cross sectional area, as it spirals from base to apex while the basilar membrane tapers in the opposite direction. On the Basilar membrane is the Organ of Corti. This, highly specialised,

structure contains hair cells, nerve endings and other, supporting, cells. Within the human cochlea there are some 15,000 hair cells which are divided between inner, and between three and five rows of outer, hair cells. The hair cells respond to deflections of the basilar membrane and cause the auditory neurons to fire, sending electrical signals to the brain. The electrical signal has two essential parts; the cochlear microphonic and the summating potential. The cochlear microphonic resembles the input speech signal while the summating potential is an offset. The outer hair cells are primarily responsible for generating the cochlear microphonic while the summating potential is generated by both the inner and outer hair cells [5]. The exact roles of the two types of cells is still not fully understood, mainly due to the lack of suitable measurements. Pickles [5], however, reviews the current understanding of cochlea mechanics.

2.2.4 The Basilar Membrane

The basilar membrane alters both in shape and tautness along its length and its frequency response thus varies accordingly. At the basal end the membrane is stiff and thin while at the apex it is compliant and massive (the ratio of stiffness is greater than 100:1). Each location along the Basilar membrane, thus has a characteristic frequency at which it will vibrate maximally. Each location can be regarded as having a constant Q bandpass filter response, meaning that the basilar membrane has best resolution at low frequencies. Also, hair cells at high characteristic frequency points respond to a wider range of frequencies than those at lower characteristic frequency points. The positions of the characteristic frequency points on the basilar membrane have distances from the apex which are approximately proportional to the logarithm of the frequency.

When a tone excites the oval window, the pressure applied to the cochlea causes the basilar membrane to vibrate at the sound's frequency. At the point on the membrane which has a matching characteristic frequency maximal vibration will occur. The auditory neurons corresponding to the hair cells at this position will then fire and electrical signals corresponding to the input tone will be sent to the brain.
Again, the exact mechanisms of the basilar membrane are still not fully understood.

While the anatomy of the ear is clear there is still much to be understood about its physiology. Further psychophysical effects are also apparent in hearing and these will now be considered.

2.2.5 Hearing Thresholds [1][3][6]

The human ear is capable of hearing sounds over a wide frequency range from approximately 16Hz to 18kHz. The exact bandwidth varies with age and possible auditory damage. In the region from 1kHz to 5kHz the ear has significantly increased sensitivity and, this range, corresponds directly with the important frequencies of speech.
The science of perception has defined a number of auditory thresholds, known as the thresholds of hearing, feeling and pain. The threshold of hearing defines the intensity of sound required before it is heard and is significantly reduced over the speech frequency range. The spectral bandpass nature of this threshold is caused by the interaction of the outer and middle ear and the concentration of hair cells at the mid-range frequencies. The two other thresholds are of less relevance to speech coding and describe the intensity at which sounds are 'felt' and at which they cause pain and possible aural damage. Typically the threshold of

pain is 120dB to 140dB (Sound intensity is measured in terms of Sound Pressure Level relative to a reference intensity of 10^{-16} watt/cm^2 at 1kHz).

Speech intensity is typically in the mid-range between the thresholds of hearing and pain. At 1m from the lips speech has a peak intensity of between 60 and 70dB.

While the threshold of hearing is relatively constant over the speech frequencies it does increase significantly below 300Hz. This effect is important since it makes low frequency reproduction quality imperative. These frequencies, contribute to the naturalness of speech and their loss contributes significantly to the 'unnaturalness' of telephone speech.

The concept of a threshold of hearing is often extended to produce curves of equal intensity for sounds. These 'equal loudness' curves describe the perceived loudness of a sound relative to its actual intensity and are used in Chapter 4 in the modelling of the ears response to sounds.

2.2.6 Masking

Masking describes the behaviour of the ear when two different sounds impinge on it simultaneously or within a short delay. One sound can simply obscure the other or one may raise the threshold of hearing of the other. The sound that becomes the masker is dependent on the circumstances, for example, when listening to a speaker at a party the speech is normally heard, however a minor distraction will cause the background noise to mask the speaker.

Masking is the most non-linear phenomenon involved in speech perception and its effects are diverse. Researchers have divided masking experiments into two classes (relevant to speech coding), with the following results:

Masking of tones by other tones

Experiments show that the masking effects around a tone are non-symmetric. Frequencies above a tone are masked more than those below it and beating effects sometimes result in increased perceptibility for tones at certain frequencies.

Masking of a tone by narrow band noise

Narrow band noise has a smoother masking effect than a tone. Close to the band the noise causes more masking than an equivalent tone but at higher frequencies the effects are similar.

These masking effects can be substantially modelled by the 'so-called' critical band effect [8], which is based on two main assumptions. The first is that when a tone is masked by noise, only those noise frequency components in a 'critical' band around it are relevant in masking. Secondly it is assumed that a tone is masked when the noise energy in the critical band equals the tone energy. The shapes of the critical bands have been extensively investigated [9] and used in modelling auditory behaviour. In Chapter 4, the concept is considered further and used as a basis for an auditory model.

2.3 Summary

This chapter has briefly described the important aspects of hearing and speech production. The discussions have concentrated on those effects that have relevance to the speech coding techniques discussed in this thesis. In particular, the psycho-acoustic effects impinge significantly on the design of low-rate speech coders. Having considered the human speech processing systems, the following chapter considers the modelling and coding of speech.

2.4 References

[1] W. H. Perkins and R. D. Kent, "Textbook of Functional Anatomy of Spech, Language and Hearing," *Taylor and Francis Ltd.*, 1986.

[2] G. Fant, "Acoustic Theory of Speech Production," *Mouton*, 1960, 1970.

[3] D. O'Shaughnessy, "Speech Communication: Human and Machine," *Addison-Wesley*, 1987.

[4] Edited by: J. Costello, "Speech Disorders in Adults," *Nfer-Nelson, Windsor*, 1985, .

[5] J. O. Pickles, "An Introduction to the Physiology of Hearing," *2nd. Ed., Academic Press*, 1988.

[6] B. C. J. Moore, "Introduction to the Psychology of Hearing," *The Macmillan Press Ltd.*, 1977.

[7] L. J. Deutsch and A. M. Richards, "Elementary Hearing Science," *University Park Press, Baltimore*, 1979.

[8] E. Zwicker, "Subdivision of the Audible Frequency Range into Critical Bands (Frequenzgruppen)," *J. Acoust. Soc. Am.*, Vol. 33, No. 2, p. 248 , Feb. 1961.

[9] E. Zwicker and E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency," *J. Acoust. Soc. Am.* , Vol 68, No. 5, pp. 1523-1525, Nov. 1980.

Chapter 3: Signal Processing Elements for Speech Coding.

Digital speech coding uses digital filters to model both the human auditory and vocal systems; this chapter considers the design of these filters, and the nature of the required excitation. Efficient representations of the filter parameters, and the excitation waveforms, are the key issues in the design of digital speech coders and this chapter discusses a number of alternative schemes. The discussion leads to the definition of a standard CELP coding scheme, which provides the basis for the work described in this thesis.

One important area of speech research is the assessment of synthesised speech quality. Since the human auditory system is highly complex, a perfect model of perception is not available, and approximate distortion measures have been developed. In this chapter, three, widely used, 'Objective' measures of coder performance are described, and the role of 'Subjective' listening tests is also, briefly, considered.

Before considering the signal processing, it is worth reviewing the form of the speech signal. The appearance of speech waveforms is a useful part of speech analysis, but it should be remembered that, since the ear's response is non-linear, a good 'visual' match does not directly equate to high 'perceived' speech quality. Figure 3.1 shows a typical utterance and three magnified sections; the aim of digital coding is to synthesise the detail of these magnified sections, while significantly reducing the quantity of information requiring transmission.

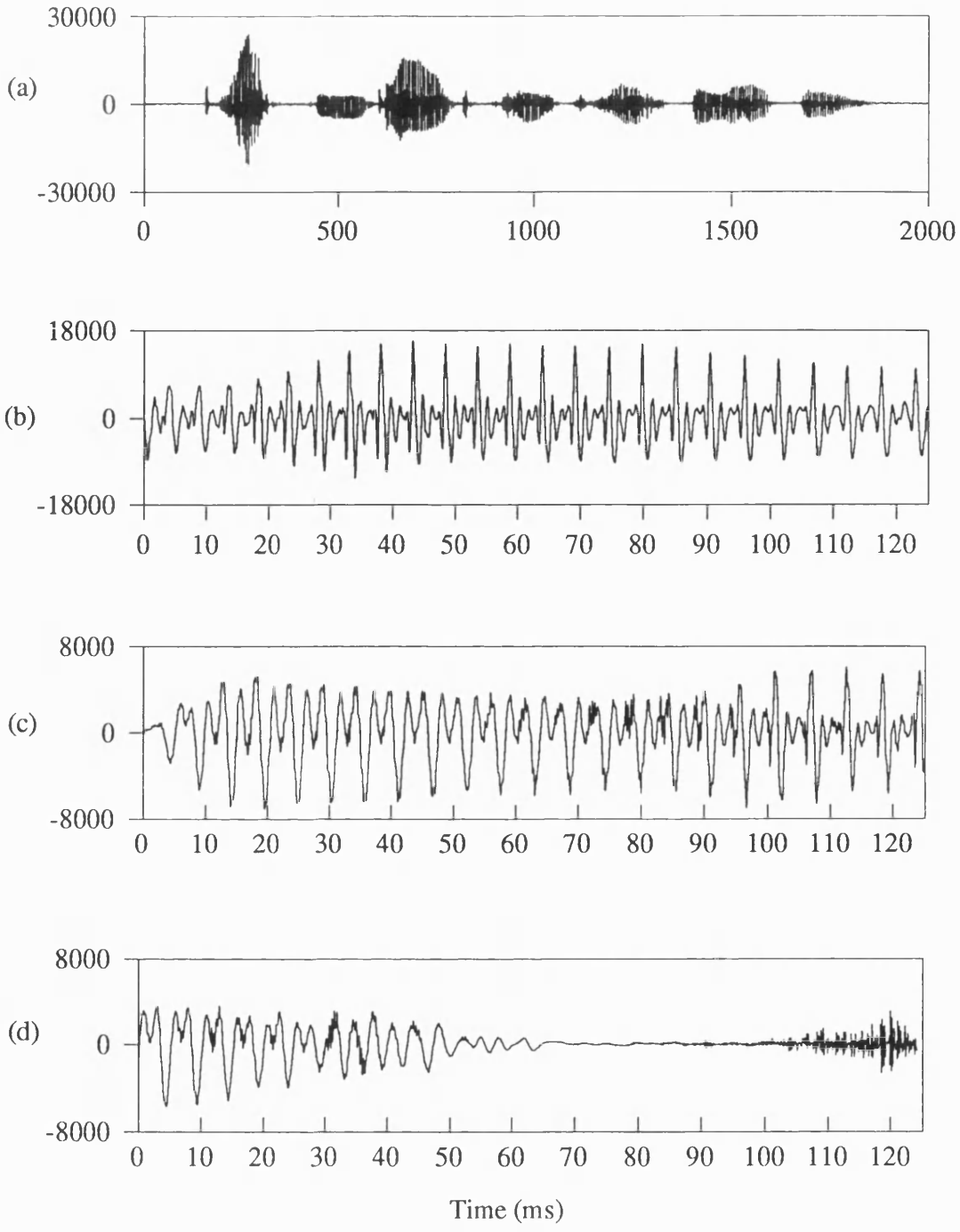


Figure 3.1: A Typical speech utterance: (a): "Cats and dogs each hate the other" by a female speaker. (b), (c) and (d): Three magnified sections of 125ms length.

3.1 A Model of Speech Production [1][2][3]

In section 2.1 the physical anatomy and basic processes of speech generation were considered, but, for coding and synthesis of speech, it is necessary to develop a simplified model of these systems. Most speech generation models can be described as the cascade of a simple time-varying linear system and an excitation generator. This simple architecture is shown in Figure 3.2.

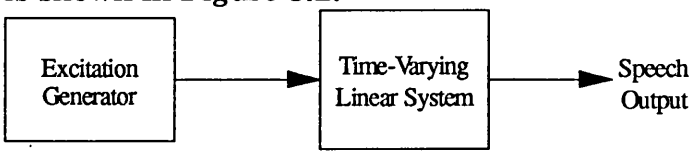


Figure 3.2: Source-system model of speech production.

In the model, the time-varying linear system simulates the vocal tract behaviour and the radiation effects of the lips while, the excitation generator either produces noise (for unvoiced speech) or a series of periodic 'glottal' pulses (for voiced speech). As discussed in section 2.1, the vocal and nasal tracts can be regarded as a collection of acoustic tubes with corresponding resonances. In the simplified model, these resonances are simulated, and altered by the parameters of the time-varying linear system.

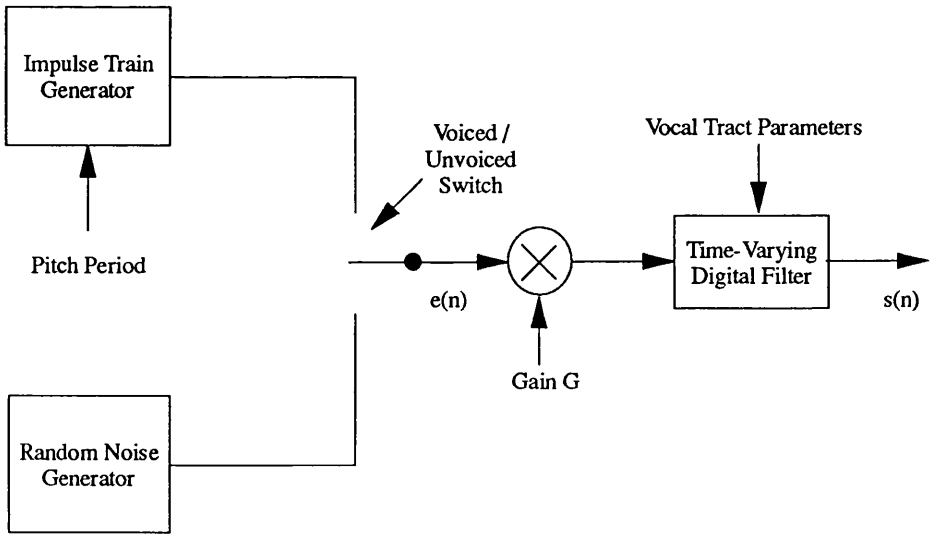


Figure 3.3: Block diagram of simplified discrete-time model for speech production.

For speech processing, the linear model needs to be defined in the discrete time domain. Rabiner [1] develops such a model, using the theory of lossless acoustic tubes, and the resulting discrete time model for speech production is shown in Figure 3.3. The nature of the time-varying digital filter, in the simplified model, will now be considered.

3.2 Linear Prediction

Linear prediction [2] is a mathematical technique by which the parameters of a linear time-varying system suitable for the architecture of Figure 3.2 can be derived. Linear prediction is used for time-series analysis and depends on the premise that a given sample of a non-random time series can be predicted as a linear combination of previous samples. Such prediction techniques can be used for many time-series ranging from physiological signals to the behaviour of share prices. Speech is quasi-periodic over short periods and is thus predictable over these lengths. Typically, linear prediction is performed over 20ms segments of sampled speech.

The P coefficients of the linear predictor are determined by minimising an error measure between the input speech and the predicted waveform. The error measure normally employed is a simple 'sum of the squared differences'.

Taking the system shown in Figure 3.3 the system function can be described by an all pole (Autoregressive) model:-

$$H(z) = \frac{S(z)}{E(z)} = \frac{G}{1 - \sum_{k=1}^{P} a(k) z^{-k}} \qquad \text{.....(3.1)}$$

where G is a gain parameter, and $a(k)$ are the prediction coefficients.

Equation 3.1 is based on the premise that speech is a predictable time series such that:

$$\tilde{s}(n) = \sum_{k=1}^{P} a(k)s(n-k) \qquad \text{.....(3.2)}$$

Thus, the error signal (or residual) between the predicted, or 'synthesised', waveform and the input speech can be expressed as:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{P} a(k)s(n-k) \qquad \text{.....(3.3)}$$

This signal can be derived, in the z-transform domain, as the output of a filter with the transfer function:

$$A(z) = 1 - \sum_{k=1}^{P} a(k)z^{-k} \qquad \text{.....(3.4)}$$

Then, if the predictor coefficients were perfect, the prediction error filter will be the inverse filter for the system *H(z)* (described in equation (3.1)) such that:

$$H(z) = \frac{G}{A(z)} \qquad \text{.....(3.5)}$$

Thus, we now have a technique whereby speech can be synthesised by passing some gain adjusted excitation through a digital filter, representing the vocal tract. The key to the Linear Prediction process, then, is to derive the predictor coefficients *a(k)*. If these can be derived and coded efficiently, speech coding and synthesis reduce to the simplified problem of representing the excitation. *e(n)*.

3.2.1 Determination of predictor coefficients

There are various recognised methods for deriving the prediction coefficients; the autocorrelation [1], covariance [1], and Burg/lattice [1][3]

techniques are just some. In this thesis the autocorrelation technique is used exclusively. This technique has gained favour in the speech coding community owing to it's low complexity and stability guarantees.

The Autocorrelation Equations

In the Autocorrelation method [1], a given speech segment s(n) (between 160 and 200 samples at 8kHz) is assumed to be identically zero outside the interval $0 \le n \le N-1$. Typically, the sequence is windowed using a Hamming window such that:

$$x(n) = s(n+N)w(n) \qquad for\ 0 \le n \le N-1 \quad .....(3.6)$$

where *w(n)* is the standard Hamming window function.

The LPC coefficients are derived by considering the error signal described by equation 3.3. In this case the total error signal energy across the predicted segment is required. This can be expressed as:

$$E = \sum_{n=-\infty}^{\infty} e^2(n) = \sum_{n=-\infty}^{\infty} \left[x(n) - \sum_{k=1}^{P} a(k)x(n-k) \right] \quad .....(3.7)$$

where *e(n)* is , as before, the residual or excitation.

To find the optimum values of *a(k)*, *E* must be minimised by setting $\partial E/\partial\, a(k) = 0$ for *k=0,1,2,3...,P*. This leads to the set of linear equations:

$$\sum_{n=-\infty}^{\infty} x(n-i)x(n) = \sum_{k=1}^{P} a(k) \sum_{n=-\infty}^{\infty} x(n-i)x(n-k) \quad for\ i = 0,1,2,3,...,P$$

.....(3.8)

This is a set of P simultaneous equations in P unknowns. Further simplification is possible by recognising that the autocorrelation of *x(n)* is defined as:

$$R(i) = \sum_{n=i}^{N} x(n)x(n-i) \qquad for\ i = 1,2,3,....,P \quad .....(3.9)$$

which reduces the simultaneous equations of equation (3.8) to:

$$\sum_{k=1}^{P} a(k)R(i-k) = R(i) \qquad for\ i = 1,2,3,....,P \qquad .....(3.10)$$

In matrix form this reduces to the solution of:

$$\begin{vmatrix} R_n(0) & R_n(1) & R_n(2) & ... & R_n(p-1) \\ R_n(1) & R_n(0) & R_n(1) & ... & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & ... & R_n(p-3) \\ ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... \\ R_n(p-1) & R_n(p-2) & R_n(p-3) & ... & R_n(0) \end{vmatrix} \begin{vmatrix} a(1) \\ a(2) \\ a(3) \\ ... \\ ... \\ a(P) \end{vmatrix} = \begin{vmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ ... \\ ... \\ R_n(P) \end{vmatrix}$$

.....(3.11)

Solution for the LPC predictor coefficients

In equation (3.11), the P by P matrix of autocorrelation values is Toeplitz (i.e. it is symmetric and all elements on a given diagonal are identically equal). This property has led to a number of efficient, recursive, solutions to the derivation of the prediction coefficients *a(k)*. Here, we shall describe only the Levinson-Durbin algorithm [1]. Other techniques, such as the Leroux-Guegen and Schur recursions [4] offer particular advantages when implemented on fixed point processors. For the work described in this thesis floating-point calculations were used exclusively, and the Levinson-Durbin procedure was sufficient.

The Levinson-Durbin recursion proceeds as follows:

$$E^{(0)} = R(0) \qquad .....(3.12)$$

$$k(i) = \left[R(i) - \sum_{j=1}^{i-1} a^{(i-1)}(j)R(i-j) \right] \Big/ E^{(i-1)} \qquad for\ 1 \le i \le P \quad .(3.13)$$

$$a^{(i)}(i) = k(i) \qquad .....(3.14)$$

$$a^{(i)}(j) = a^{(i-1)}(j) - k(i)a^{(i-1)}(i-j) \qquad for\ 1 \le j \le i-1 \quad (3.15)$$

$$E^{(i)} = (1-[k(i)]^2)E^{(i-1)} \quad \text{.....(3.16)}$$

The final solution for the prediction coefficients is then given as:

$$a(j) = a^{(P)}(j) \quad \text{.....(3.17)}$$

In the process of solving for the prediction coefficients the solutions for all lesser order predictors are also found. As the predictor order is increased, the error signal resulting from the LPC filter will decrease in magnitude. The results of other authors [5] show that for voiced speech, once there are enough poles to model the formant structure additional poles do little to reduce the error; this 'critical' order is normally around P=12. However, in practice, most current speech coding techniques compromise and use 10th order LPC for 8kHz sampled speech.

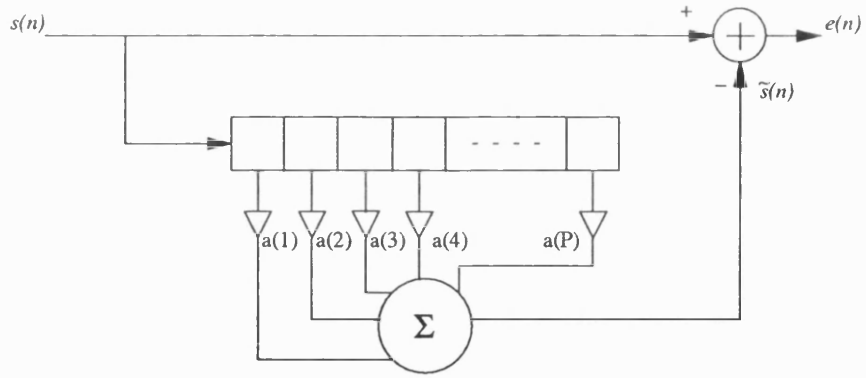
Further, it is possible to make a simple check for the stability of the solution by monitoring the parameters *k(i)*. It is necessary and sufficient that the roots of the polynomial *A(z)* are inside the unit circle for the system H(z) to be stable [1][3]. This produces a condition on the *k(i)* such that they satisfy:

$$-1 \le k(i) \le 1 \quad \text{.....(3.18)}$$

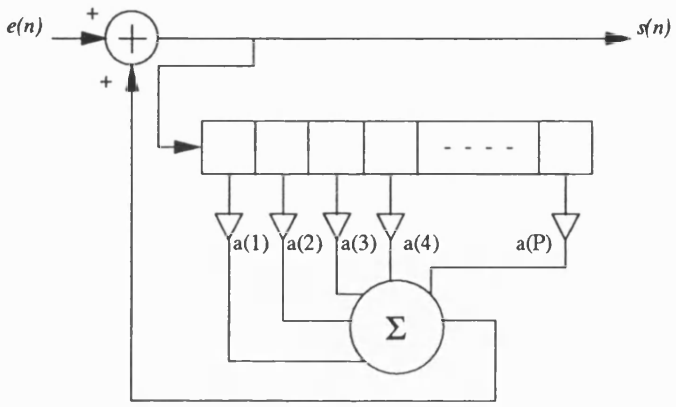
The *k(i)* are known as the PARtial CORrelation (or PARCOR) coefficients and their negatives as the reflection coefficients. These can be related to parameters of acoustic tube models [1].

3.2.2 The LPC Analysis and Synthesis Filters

The simplest LPC analysis filter (i.e. a filter which generates the residual from input speech) would be a direct FIR implementation as shown in Figure 3.4(a). Note that, while the analysis filter is FIR, the synthesis (or Inverse LPC) filter (3.4(b)) is IIR. The nature of the LPC residual produced by an analysis filter is shown in Figure 3.5.



(a)



(b)

Figure 3.4: Structures of (a) Direct form LPC Analysis Filter and (b) Direct Form LPC Synthesis Filter.

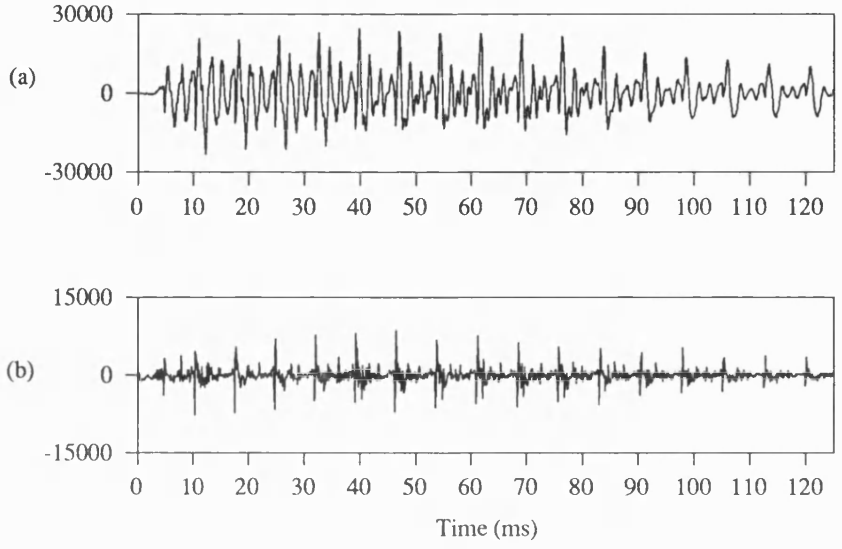


Figure 3.5: The LPC residual (b) produced by an LPC Analysis Filter from the input speech record (a).

The direct form structure is, however, very sensitive to small prediction coefficient errors, such as those generated by quantisation and channel errors. It is thus useful to consider an alternative lattice filter architecture, which has better properties for quantised parameters since it uses the reflection coefficients *-k(i)* as multipliers. These are less spectrally sensitive to quantisation than the prediction coefficients *a(i)* [1]. Lattice filter structures are shown in Figure 3.6 and the LPC filters used in this thesis are based on this scheme.

While the reflection coefficients are less spectrally sensitive to quantisation than the predictor coefficients, they are still not suitable for direct transmission at low bit rates over high error rate radio channels. In these environments, the relatively small quantisation/channel errors can still result in significant distortion of the output speech. Thus, a number of more robust coded forms for the LPC parameters have been suggested. One encoding technique for the reflection coefficients is the Log Area

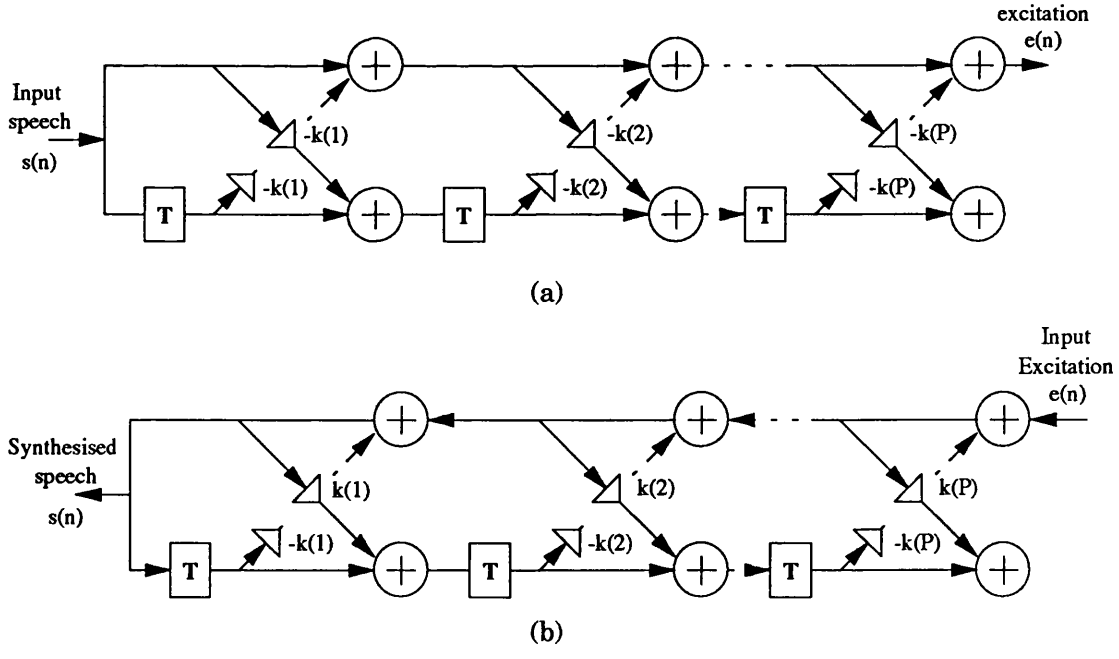


Figure 3.6: LPC Lattice Filter Structures: (a) Analysis filter, and (b) Synthesis Filter.

Ratio (LAR) [1]. In this representation the reflection coefficients *k(i)* are transformed according to:

$$g(i) = \log\left[\frac{1-k(i)}{1+k(i)}\right] \qquad for\ 1 \le i \le P \qquad \text{.....(3.19)}$$

The LARs are particularly appropriate since the non-linear transformation expands the critical region about $|k(i)| = 1$. This allows uniform quantisation to be used with decreased spectral sensitivity.

While the Log Area Ratios are a significant encoding improvement, for low bit rate coders (sub 8kbit/s) a more efficient technique is required. In 1984 Itakura [6] introduced Line Spectral Pairs (or frequencies). These have been found to have particularly good quantisation properties. A recent scheme, using Line Spectral Frequencies, codes the LPC parameters for a 160 sample frame of 8kHz sampled speech using just 25 bits. The derivation of Line Spectral Frequencies is now, briefly, considered.

3.2.3 Line Spectral Frequencies

In the Linear Predictive analysis, described in section 3.2, a short speech segment is assumed to be generated as the output of an all-pole filter with suitable excitation. Ignoring the gain as being included in the excitation source, the all-pole synthesis filter is $H(z) = 1/A(z)$ where *A(z)* is given by:

$$A(z) = 1 + a(1)z^{-1} + a(2)z^{-2} + a(3)z^{-3} + ...a(P)z^{-P} \qquad \text{.....(3.20)}$$

To define the Line Spectral Frequencies (LSFs) the inverse filter polynomial is split into two new polynomials:

$$P(z) = A(z) + z^{-(P+1)}A(z^{-1}) \qquad \text{.....(3.21)}$$

$$Q(z) = A(z) - z^{-(P+1)}A(z^{-1}) \qquad \text{.....(3.22)}$$

It directly follows that:

$$A(z) = [P(z) + Q(z)]/2 \qquad \text{.....(3.23)}$$

The roots of the sum and difference equations ((3.21) and (3.22) respectively) are called the Line Spectral Frequencies. The LSFs benefit from two properties of the roots of these two equations:

1. All zeros of P(z) and Q(z) lie on the unit circle.
2. Zeros of P(z) and Q(z) are interleaved, hence making the LSFs interleaved.

The second property means that the LSFs are taken in ascending, alternating order as the roots of P(z) and Q(z). Such properties clearly make the calculation of the LSFs simpler, since roots need only be searched for from the lower limit of the previous root. Further, if the LSFs do not satisfy these simple criteria then the associated LPC synthesis filter is unstable. The tracks of the first ten LSFs of a typical speech segment are shown in Figure 3.7, where the interleaved nature of the LSFs can be clearly seen. Further, the 'clustering' of the LSFs indicate formants or, at least, spectral peaks of the input speech [7].

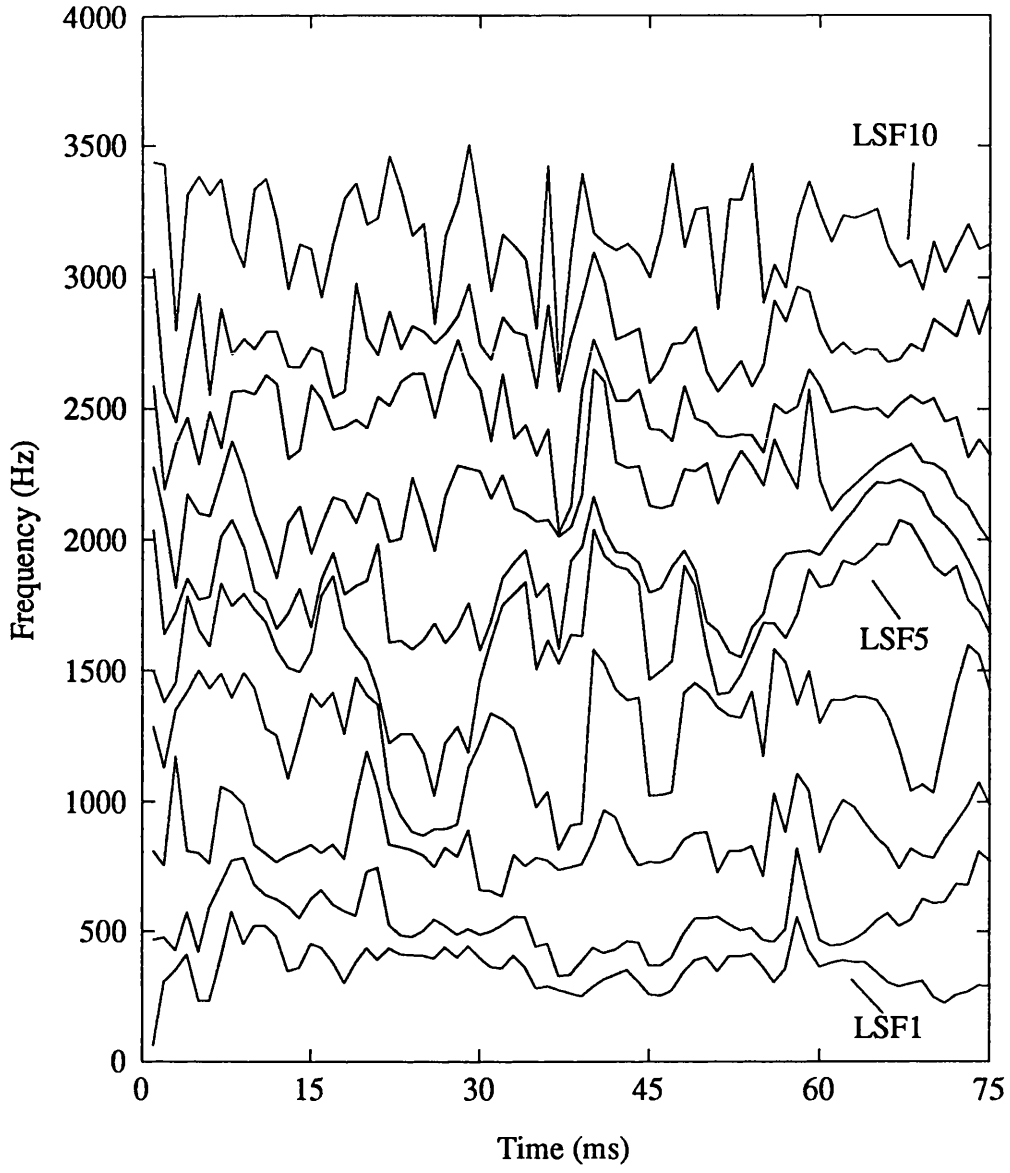


Figure 3.7: The tracks of the first 10 LSFs for the utterance 'Mechanical Ingenuity has ....' by a male speaker.

The computation algorithms used for LSFs are beyond the scope of this thesis and are detailed extensively in the references [8][9]. The procedures have developed extensively over the last twelve years since the LSF technique was introduced and, in particular, the work of Kang and Fransen [8] suggests two techniques. In the early work of this thesis the 'Approach 1: Using the Amplitude Response of the Sum and Difference filters', from [8] was used to derive the LSFs. However, in a

practical coder this technique is too time consuming and the later work (particularly that of Chapter 5) used the technique employed by the US Federal Standard 1016 speech coder. This is based on the technique described by Kabal and Ramachandran [9], which, for solution purposes, casts the sum and difference equations as Chebyshev polynomials.

A simple technique, suggested by Kang and Fransen [8], was used to perform the inverse transformation of LSFs to prediction coefficients. This uses a direct LSF form of the LPC analysis filter, the impulse response of which provides the LPC coefficients. The architecture of this filter is shown in Figure 3.8.

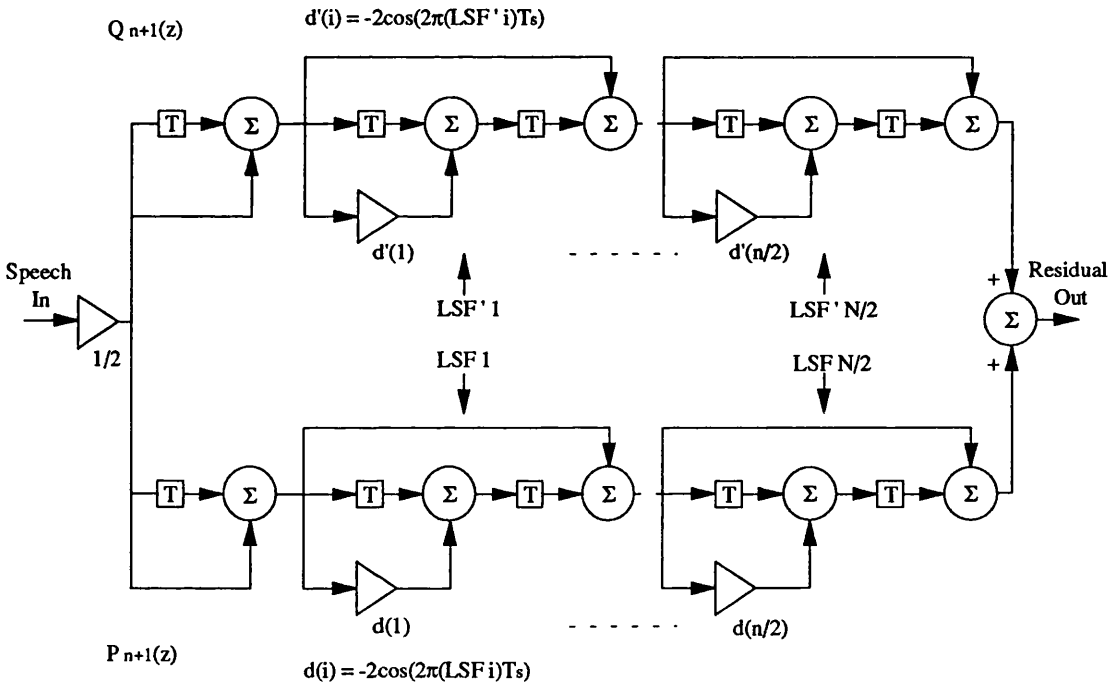


Figure 3.8: Block Diagram of the LPC Analysis Filter configuration, using LSFs as filter weights.

3.3 Long Term or Pitch Prediction

The LPC filter, described in the previous section, removes the spectral envelope of input speech. However, in the previous sections on speech generation it was clear that speech also contains important, long term pitch information below 500Hz. For this reason, in the model of Figure 3.3, a voiced/unvoiced switch, switching between a pitch-periodic or noise-like excitation source, is included. Many current speech coders, though,

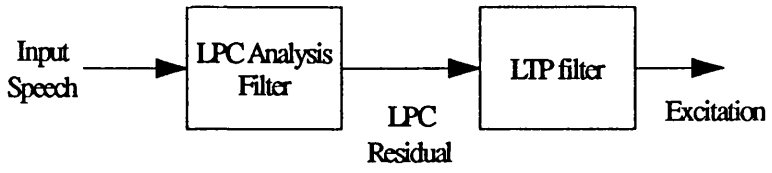


Figure 3.9: Cascade of LPC Analysis Filter and Long Term Predictor (LTP)

contain no such switching, and a source of periodicity is always included in the excitation. For this purpose, a Long Term Predictor (or LTP) [10] is cascaded with the short term LPC filter; a typical analysis structure is shown in Figure 3.9

The LTP removes long term (or pitch) correlations from the LPC residual. This is performed by subtracting a prior, delayed section of residual which maximally correlates with the current filtered section. The delayed section is also optimally gain adjusted, such that the parameters for the LTP are a delay and gain term. The LTP filtering process is shown

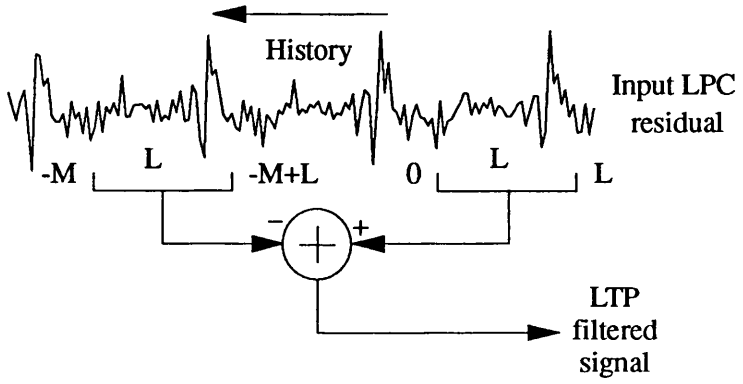


Figure 3.10: The operation of an LTP 'analysis' filter.

diagrammatically in Figure 3.10 and the delay and gain computations are now described mathematically. Note that the LTP filtering process is identically reversible for most delays.

For a segment length of L, an input residual r(n), and a candidate vector f(n), the squared prediction error between the two segments will be:

$$E = \sum_{n=0}^{L-1} [r(n) - f(n)]^2 \qquad \text{.....(3.24)}$$

For a LTP, however, the candidate vector is a section from the 'history' of the residual. If the candidate vector is assumed to commence at a delay of M samples then equation 3.24 becomes:

$$E(M,\lambda) = \sum_{n=0}^{L-1} [r(n) - \lambda r(n-M)]^2 \qquad \text{.....(3.25)}$$

where λ is an optimum gain term, which is determined by setting the derivative $\partial E(M,\lambda)/\partial\lambda = 0$, leading to:

$$\lambda = \frac{\sum_{n=0}^{L-1} r(n)r(n-M)}{\sum_{n=0}^{L-1} r^2(n-M)} \qquad \text{.....(3.26)}$$

Substituting for λ in equation 3.25 leads to the error function:

$$E(M) = \sum_{n=0}^{L-1} r^2(n) - E'(M)$$

$$\text{where } E'(M) = \frac{\left[\sum_{n=0}^{L-1} r(n)r(n-M)\right]^2}{\sum_{n=0}^{L-1} r^2(n-M)}$$

.....(3.27)

The function $E'(M)$, alone, can be used for the search by noting that a squared error will never be negative. Thus, the search proceeds for all candidate vectors and the one which maximises $E'(M)$ is chosen as the optimum candidate at delay M. For a periodic signal, the delay will correspond with the pitch period delay or multiples thereof, while non periodic (or unvoiced) signals have more erratic behaviour. Typically, pitch delays are searched over a range of between 16 and 147 samples at 8kHz, corresponding with pitch frequencies of 500Hz and 54Hz, respectively.

The above analysis is for a first order predictor, but other authors have used multiple-tap filters [11] and non-integer delay filters [12][13]. The latter, interpolate between the samples by use of bandpass interpolation [14]. Both techniques offer performance advantages, however the non-integer methods show the greatest promise and a simplified, non-integer approach is specified in the US Federal Standard 1016 coder [15][16][17]. In this thesis, integer-delay pitch prediction is used; this reduces coder complexity while still giving valid results. Improved performance for the coders described could, however, be achieved by use of non-integer delay techniques.

The type of LTP described in this section is known as an 'open-loop LTP' for reasons that will become apparent in the following sections. Typical results of the cascade of a first-order open-loop LTP and an LPC analysis are shown in Figure 3.11

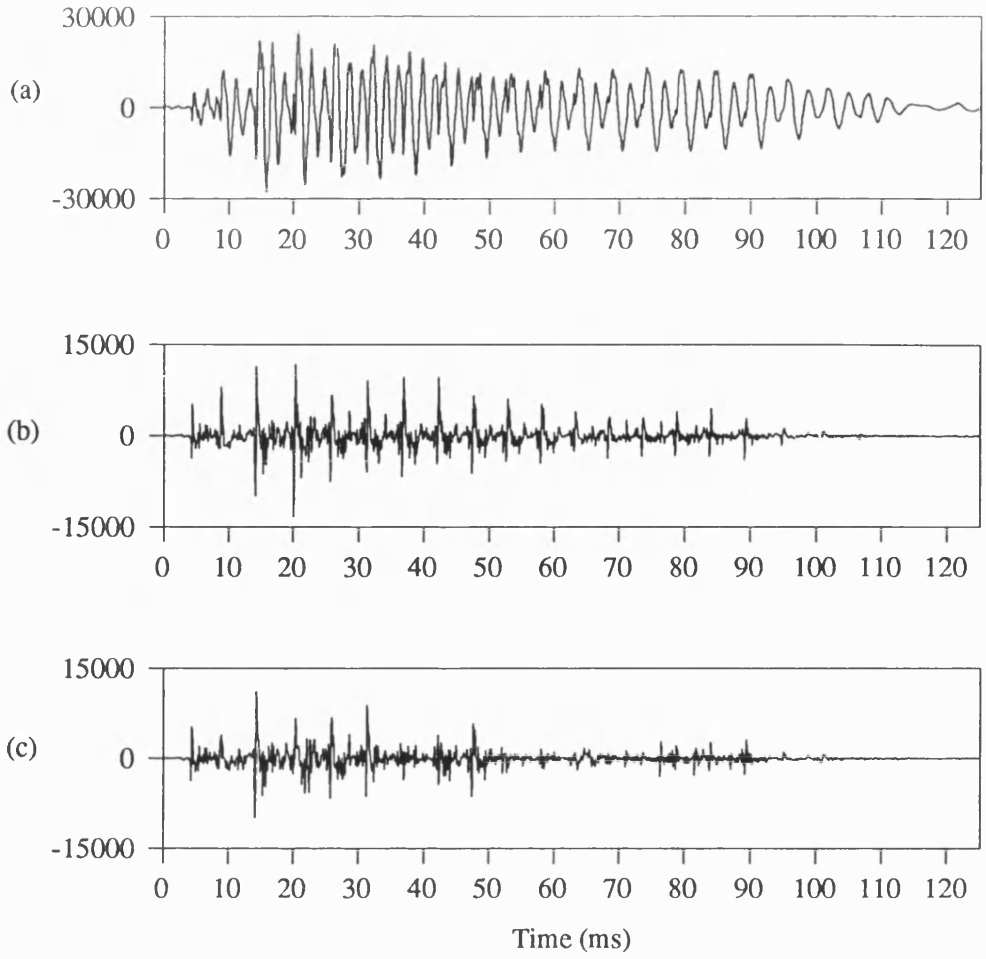


Figure 3.11: Waveforms showing operation of an open-loop LTP: (a) Input Speech (b) LPC Residual and (c) Excitation output of first-order LTP.

3.4 Analysis-by-Synthesis Coding

While direct coding of the excitation is possible, such an approach is sub-optimal and does not make full use of the available bits. The full rate GSM speech coder [18] quantises the excitation as a stream of regularly spaced pulses in an 'open-loop' manner. This requires coding of position and gain terms for the excitation and requires high bit rates (the GSM coder operates at 13kbit/s). For rates of sub 8kbit/s it is necessary to code the excitation more efficiently. This can be achieved by 'closed-loop' analysis-by-synthesis architectures, an example of which is shown in Figure 3.12.

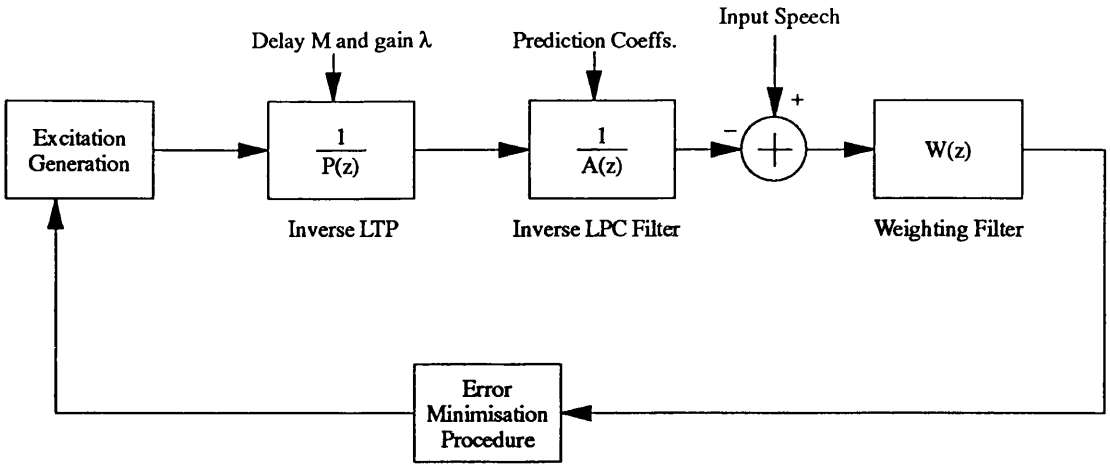


Figure 3.12: A general Analysis-by-Synthesis speech coding architecture (with open-loop LTP).

Ignoring the weighting filter, the process can be regarded as choosing an excitation vector which, when filtered by the LTP/LPC cascade best matches the input speech segment. The procedure is normally applied on a block-by-block basis where each speech segment is 5 to 10ms in duration. At a sampling rate of 8kHz, this corresponds to 40 to 80 samples. Since the LPC coefficients are normally calculated for a frame length of 160 to 240 samples the frame is divided into sub-frames for the analysis-by-synthesis excitation vector search. Generally a frame is divided into four sub-frames.

Overall, the Analysis-by-Synthesis (A-by-S) approach allows the best possible excitation from the available selection to be chosen. Unlike parametric excitation representations, the excitation is chosen on the basis of the synthesised speech it produces. This is clearly a more appropriate approach if the increased computational complexity can be tolerated.

3.4.1 Excitation Representations

Early A-by-S architectures used Multi-pulse Excitation (MPE) [19], which represents the excitation as a series of pulses located at non-uniformly

spaced intervals. The coder has control over both the position and amplitude of each pulse. Typically, there are 4 or 5 pulses per 5ms segment making simultaneous optimisation of all pulses highly complex, and sub-optimal, sequential solutions are normally adopted. These operate by optimising each pulse position and amplitude in turn. After each pulse is determined its contribution to the synthesised speech is taken into account during the remaining pulse optimisations.

A simplification of the multipulse representation is known as Regular Pulse Excitation (RPE) [10]. Such coders, represent the excitation as a series of regularly spaced pulses. The coder optimises only the position of the initial pulse and the amplitude of each pulse.

Typical multi-pulse and regular pulse representations are shown in Figure 3.13. While these parametric excitation schemes can offer good speech quality at bit rates of around 13kbit/s, they do not offer a suitable excitation representation for coders operating below 8kbit/s.

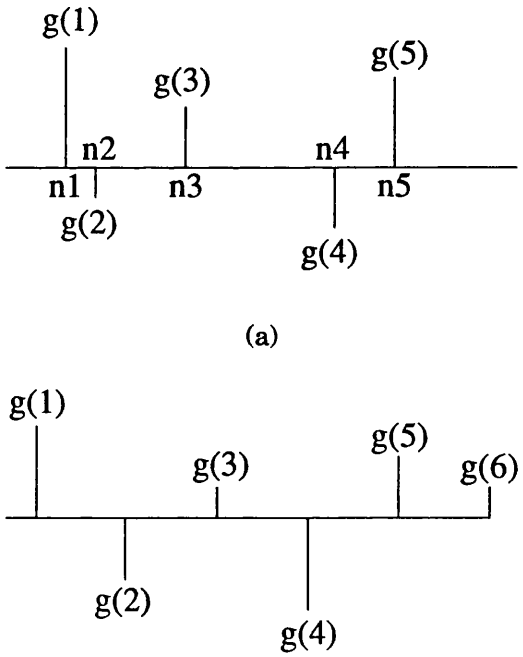


Figure 3.13: A typical (a) Multi-pulse excitation (MPE) vector, and (b) Regular Pulse Excitation (RPE) vector. Each vector would represent a single 5ms sub-frame.

Perhaps the most important excitation representation to be suggested in recent years is Code Excited Linear Prediction. In [21], Schroeder and Atal suggested that the excitation could be represented by a codebook of Gaussian vectors. (A Gaussian vector, or sequence is a series of Gaussian sample values). Examples of Gaussian sequences are shown in Figure 3.14.

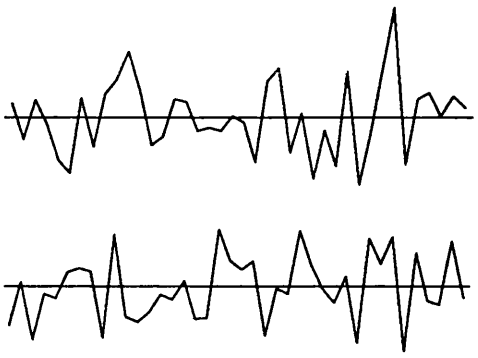


Figure 3.14: Examples of Gaussian codebook sequences.

Schroeder and Atal found that good speech could be synthesised by forming the Inverse filter input as the output of an LTP in conjunction with a Gaussian codebook search. The major advantage of such a scheme is the low number of bits required to represent the sub-frame excitation e.g., for a typical 1024 vector codebook just 10 bits are needed to represent the excitation 'shape'. The required gain parameter can be quantised, coarsely, using a 5-bit, 32 level non-linear quantisation scheme.

While all the excitation schemes presented can produce good quality output speech, it is important that the error criterion, used for vector choice, is optimised to fully exploit the behaviour of the human auditory system. The role of the weighting filter, in Figure 3.12, in such a criterion is now considered.

3.4.2 A Perceptual Error Criterion - the Weighting Filter

The A-by-S architecture, shown in Figure 3.12, minimises the error between the sub-frames of input speech *s(n)* and synthesised speech $\tilde{s}(n)$. This error is commonly calculated as the mean-squared error between the vectors, however for high quality speech synthesis it is necessary to incorporate auditory masking effects.

In section (2.2.6), the masking behaviour of the human ear was discussed. Simply, the ear has only limited ability to perceive small errors in frequency bands where there is high energy (i.e. around speech formants). To use this effect it is necessary to redistribute the speech power away from the formants, such that these bands are de-emphasised and more critical, low energy, bands are emphasised. This is the role of the weighting filter; Figure 3.15 compares typical speech and weighting filter spectra.

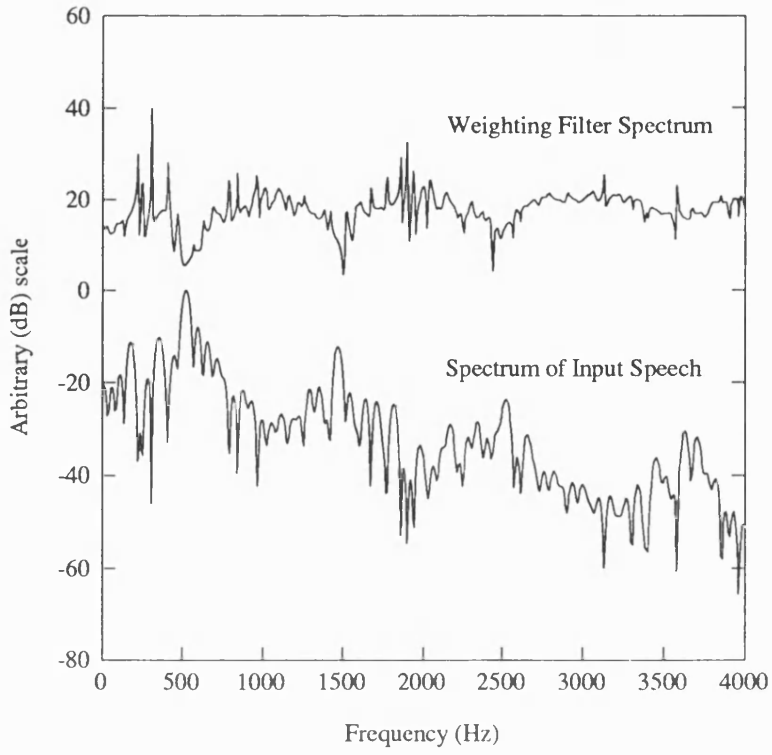


Figure 3.15: Comparison of typical speech spectrum and that of the corresponding Weighting Filter. Transform length is 256 samples.

The weighting filter (or noise shaping) filter is derived from those used in Adaptive Predictive Coders (APC) [21], where a suitable filter characteristic for speech was found to be:

$$W(z)=\frac{A(z)}{A(z/\gamma)}=\frac{1-\sum_{i=1}^{P}a(i)z^{-i}}{1-\sum_{i=1}^{P}a(i)\gamma^{i}z^{-i}} \qquad for 0\le\gamma\le 1 \qquad .....(3.28)$$

where $A(z)$ is the standard LPC analysis filter and the parameter γ is normally given a value of 0.8 or 0.9. γ controls the energy in the formant regions, such that decreasing γ, de-emphasises the formants by increasing the bandwidths of the poles of $W(z)$. The increase in bandwidth $\Delta\omega$ is given by [10]:

$$\Delta\omega=\frac{-f_s}{\pi}\ln\gamma \text{ Hz} \qquad .....(3.29)$$

In equation (3.29), f_s is the sampling frequency, in this case 8kHz, and, for a value of γ $=0.9$, the approximate bandwidth increase is 250Hz.

In the A-by-S architecture it is now possible, after making some assumptions, to considerably reduce the computational load of the search. If it is assumed that the IIR synthesis filter decays sufficiently within the sub-frame to be considered FIR, then the weighting and synthesis filters can be cascaded to produce a new weighted synthesis filter:

$$H_{\gamma}(z)=\frac{1}{A(z)}\frac{A(z)}{A(z/\gamma)}=\frac{1}{A(z/\gamma)} \qquad .....(3.30)$$

This new, 'weighted' filter will have an impulse response similar to that of the standard LPC synthesis filter, but it will decay rapidly due to the influence of the γ term.

The new weighted impulse response will be related to the original *h(n)* as:

$$h_\gamma(n) = \gamma^n h(n) \qquad n = 0,1,2,..... \qquad .....(3.31)$$

The weighted synthesis filter is implemented in both the direct and lattice forms by inserting a multiplier γ before each delay element.

The incorporation of the weighting operation into the synthesis arm of the A-by-S system requires that the input speech is now weighted by *W(z)* prior to the error minimisation. Thus the revised A-by-S CELP architecture is shown in Figure 3.16. Future discussions exclusively consider the codebook excitation technique and CELP.

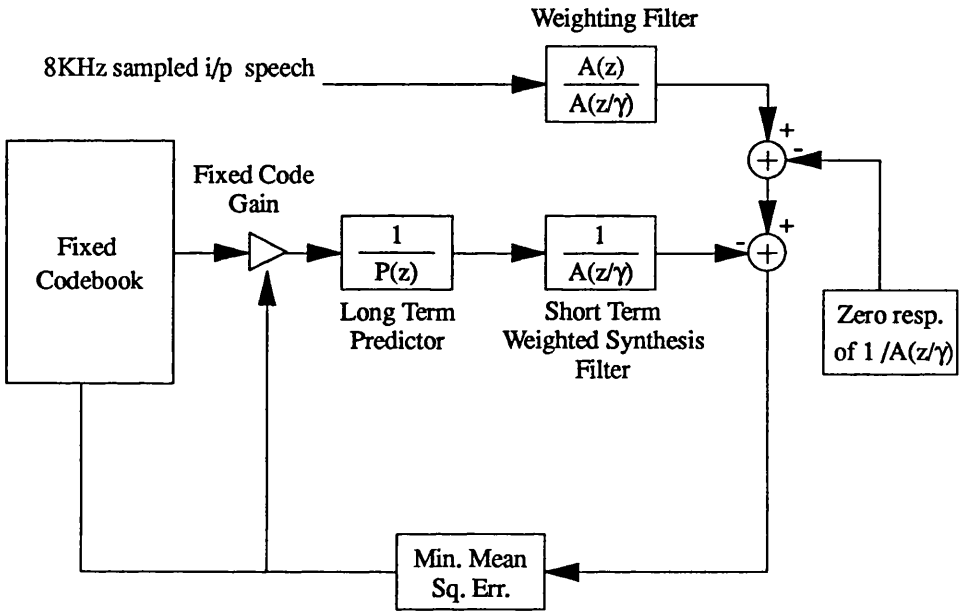


Figure 3.16: The Time Domain A-by-S CELP Architecture (with LTP).

3.4.3 Error Minimisation Procedure for codebook search

Before detailing the error minimisation it is worth simplifying the problem. It is convenient to consider the cascaded LTP and LPC weighted synthesis filters (see Figure 3.16) to be a single filter H(z). The error search is now described in the convenient matrix/vector terminology. In this case **H** will be an L by L lower triangular matrix constructed from the truncated impulse response of H.

The impulse response is truncated at the sub-frame length, L, such that **H** is described by:

$$\mathbf{H}=\begin{vmatrix} h(0) & 0 & 0 & \dots & 0 \\ h(1) & h(0) & 0 & \dots & 0 \\ h(2) & h(1) & h(0) & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ h(L-1) & h(L-2) & h(L-3) & \dots & h(0) \end{vmatrix} \qquad \text{.....(3.32)}$$

The synthesised speech vector will then be described by the convolution of the code vector $\mathbf{c^{(q)}}$ and the weighted filter response such that:

$$\tilde{\mathbf{S}} = \mathbf{Hc^{(q)}} \qquad \text{.....(3.33)}$$

Then the total mean squared error between the synthesised speech and the input , weighted speech, will be:

$$E^{(q)} = \left\| \mathbf{s_w} - \chi^{(q)}\mathbf{Hc^{(q)}} \right\|^2 \qquad \text{.....(3.34)}$$

where χ is the code gain term and $\| \ \|^2$, the sum of the squares of the vector components. Putting $\partial E^{(q)}/\partial \chi^{(q)} = 0$ in equation (3.34), a solution for the scalar χ can be found, such that:

$$\chi^{(q)} = \frac{\mathbf{s_w^T Hc^{(q)}}}{\left\| \mathbf{Hc^{(q)}} \right\|^2} \qquad \text{.....(3.35)}$$

Now, substituting for χ in (3.34) an expression for the squared error for the qth codebook entry is found:

$$E^{(q)} = \|\mathbf{s}\|^2 - \frac{\left[\mathbf{s^T Hc^{(q)}}\right]^2}{\left\| \mathbf{Hc^{(q)}} \right\|^2} \qquad \text{.....(3.36)}$$

This can be further simplified by noting that a squared error will never be negative. Hence the expression of (3.36) reduces to the maximisation across the codebook of :

$$E'^{(q)} = \frac{\left[\mathbf{s}^{T}\mathbf{H}\mathbf{c}^{(q)}\right]^{2}}{\left\|\mathbf{H}\mathbf{c}^{(q)}\right\|^{2}} \qquad \text{.....(3.37)}$$

Thus, to summarise, the CELP search finds the weighted squared error between the input and synthesised speech for every excitation vector in the codebook. The codebook entry q which maximises (3.37) is chosen to represent the current sub-frame. The codebook index q and the quantised gain term $\chi^{(q)}$ are then transmitted as the excitation representation for the current sub-frame. It should be noted that a CELP receiver structure is simply a codebook, and cascade of the inverse LTP and the unweighted LPC synthesis filter, $1/A(z)$.

One further complication is caused by the truncation of the IIR synthesis filter in the CELP search. This filter is IIR and even if the current sub-frame had an identically zero excitation the filter would produce an output. It is necessary to take this 'zero-response' into account during the search process, and, thus, prior to the CELP search, the 'zero-response' of the synthesis filter is subtracted from the weighted input speech. This modification is included in the CELP architecture shown in Figure 3.16.

3.4.4 A 'closed-loop' LTP - an adaptive codebook.

While the excitation in the CELP coder described is chosen in a 'closed-loop' optimisation, the LTP parameters must still be computed 'open-loop'. Normally the 'open-loop' LTP parameters are computed from the input speech sub-frame. It has, however, been shown [22] that significant performance improvements can be achieved by making the LTP part of

the CELP optimisation loop. In this way, the LTP contribution to the residual will interact in an optimal manner with the excitation.

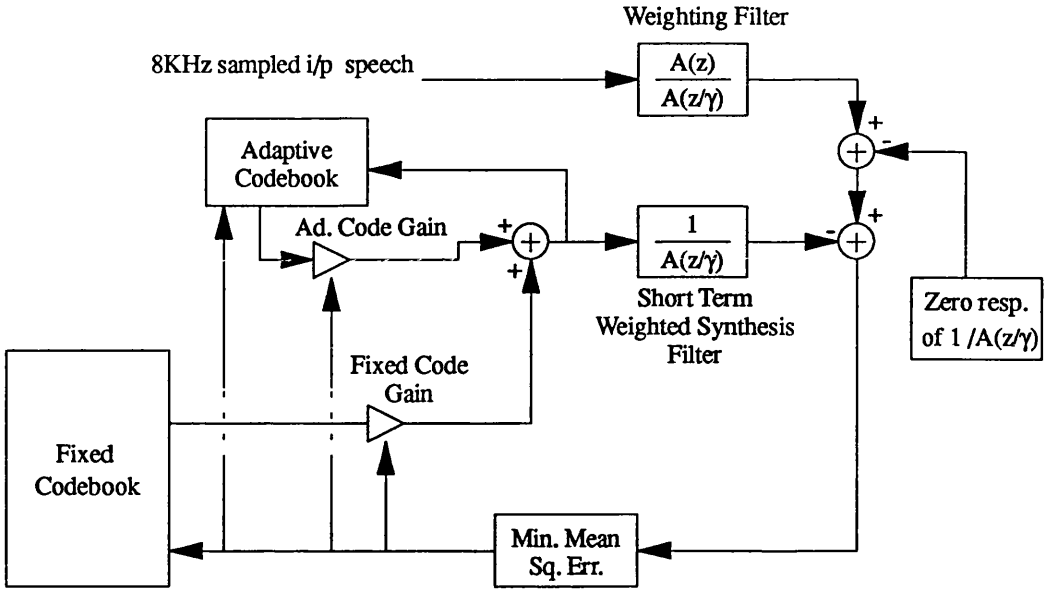


Figure 3.17: The standard Time Domain A-by-S CELP Architecture.

So as to include the LTP in the CELP loop it is necessary to recast the CELP architecture. Figure 3.17 shows the new architecture, where the LTP is removed and replaced by an adaptive codebook search. The entries of this codebook are generated from the previous 'history' of the residual. The adaptive codebook can, thus, be regarded as a shift register. The adaptive codebook concept is shown in Figure 3.17.

Essentially, the adaptive codebook is a set of 'overlapped' codebook vectors. Each vector of index (-d), overlaps the vector of index (-d+1) by all but the last sample with a new value for its first sample.

Each code is derived such that:

$$a^{(d)}(i) = r(n-d+i) \qquad for\ 0 \le i < L\ and\ d \ge L \ldots..(3.38)$$

The codebook search proceeds identically to that described for the excitation codebook in the previous section (the excitation codebook is generally referred to as the 'Fixed' codebook, to avoid confusion). The adaptive codebook parameters transmitted correspond directly with those

of a LTP and are the chosen adaptive code index, d, and the respective gain term χ_a. Adaptive codebooks are normally searched over 128 codes for delays between 16 and 143.

The alteration in the CELP architecture also simplifies the search convolutions since the LTP response is no longer present. Thus in the error minimisation, the matrix **H** is now constructed with the impulse response of the LPC weighted synthesis filter $1/A(z/\gamma)$.

There is, however, one problem with the adaptive codebook structure. Codes that have delays of less than the sub-length (i.e. 40 samples for a 5ms sub-frame) are incompletely defined. While the required samples can be found recursively, this is impractical in a codebook search. The approximation used in the work for this thesis [23], periodically extends such codes. The adaptive code vectors, $a^d(n)$, for $d<L$ are then defined as:

$$\begin{aligned} a^{(d)}(i) &= r(n-d+i) && \text{for } 0 \le i < d \\ a^{(d)}(i) &= a^{(d)}(i-d) && \text{for } d \le i < L \text{ and } d < L \end{aligned} \qquad \text{.....(3.39)}$$

The overlapped nature of the adaptive codebook can also be used to reduce search complexity [23]. The technique can also be applied to the fixed codebook and this is now considered.

3.4.5 Fixed Codebook Implementation

The CELP fixed codebook could be directly implemented as a full codebook consisting of, typically, 1024 40-sample vectors. Such a codebook requires 163,844 bytes (approx. 160Kbytes) if all sample values are held in single-precision floating point. This is a considerable fixed memory requirement for a speech coder which may operate in a portable environment.

An alternative codebook approach is to calculate each code on-line from a seeded Gaussian sequence generator. Such an approach is practical for

simulation but in real-time coders the CELP search is on the limits of current processing capability [16]. The inclusion of the generator would add a significant processing overhead making a real-time CELP search impractical.

The accepted solution to the fixed codebook implementation is an 'overlapped' codebook, which is similar to the adaptive codebook described previously. An overlapped fixed codebook consists of a long Gaussian sequence (for a 1024, 40 sample vector codebook: 1024*2+40=2088 samples). Lin [24] found that a two sample code overlap offered performance advantages over the single sample overlap of the adaptive codebook. For the fixed codebook the codes are generated as:

$$f(q,n) = G(2q+n) \qquad n = 0,1,2,....,40 \qquad .....(3.40)$$

where $G(n)$ is a unit variance Gaussian sequence.

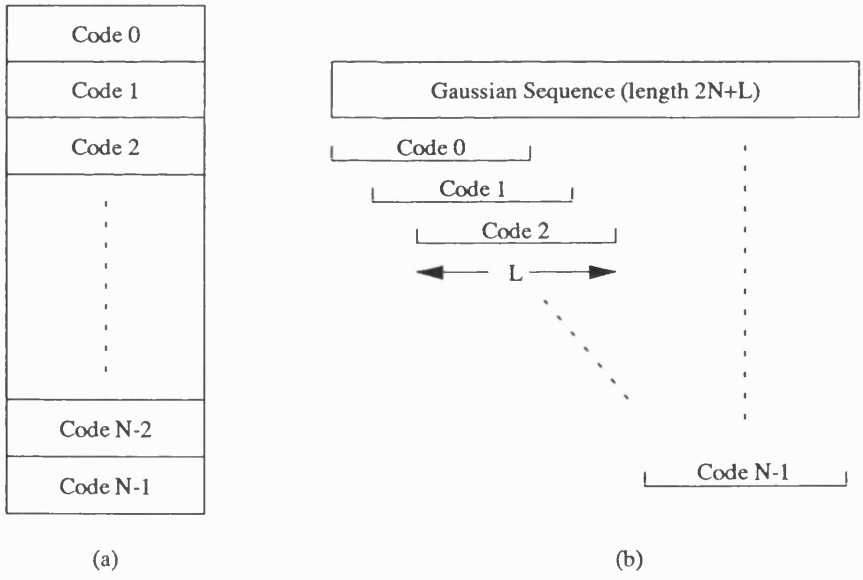


Figure 3.18: The structure of (a) a Full fixed codebook and (b) an overlapped fixed codebook.

A comparison of the Full fixed and overlapped codebook structures is shown in Figure 3.18. The overlapped nature of the codebook allows a considerable simplification of the CELP search. In the CELP search

successive codes are convolved with an identical inverse LPC filter response. In the overlapped codebook the result of this convolution for each new code will be identical to that for the previous vector excepting two samples and a shift. Thus rather than perform a convolution requiring $L^2/2$ multiplications, an end correction procedure requiring just *4L* multiplications is performed. The 'end-correction' proceeds as follows:

1.Subtract previous code's first sample contribution from convolution result:

$$\tilde{s}(i) = \tilde{s}(i) - G(2q-2)h(i) \quad 0 < i < L \qquad \text{.....(3.41)}$$

2.Shift previous code's convolution result, dropping first sample:

$$\tilde{s}(i) = \tilde{s}(i+1) \qquad 0 < i < L-1 \qquad \text{.....(3.42)}$$

3.Add contributions of new code's last sample to convolution result:

$$\tilde{s}(L-1) = \sum_{i=0}^{L-1} G(2q-2+i)h(L-1-i) \qquad \text{.....(3.43)}$$

This process is repeated, in the case of an overlap of 2, with *2q-1* replacing *2q-2* in equations (3.41)-(3.43). This scheme can also be used to improve the adaptive codebook search.

Further adaptations and derivations of the of the overlapped codebook technique have been suggested by a number of authors [25][26]. Ternary codebooks [26] are of particular interest since the Gaussian sequence is replaced by a centre clipped series, resulting in a series of signed, unit impulses. These irregularly spaced pulses, interspersed with zero values, can be stored efficiently and a further reduction in computational complexity is produced by the multiplications becoming a limited number of additions. An overlapped ternary codebook has been adopted for the Federal Standard 1016 speech coder [17].

3.5 A Standard Time-Domain CELP Architecture

In the previous sections of this chapter we have fully described the signal processing elements of the CELP A-by-S speech coder. In this section the

	Prev. Frame LSF	Pres. Frame LSF
Sub-frame 1	0.75	0.25
Sub-frame 2	0.5	0.5
Sub-frame 3	0.0	1.0
Sub-frame 4	0.0	1.0

Table 3.1: Interpolation proportions for LSFs in standard Time Domain CELP Coder.

parameters of a standard CELP architecture will be defined. The results of this coder are used as a benchmark for the coders developed in the following chapters of this thesis.

The standard Time Domain CELP architecture is identical to that shown in Figure 3.17. The coder operates on 8kHz sampled speech and a frame length of 160 samples (20ms). The LPC parameters are determined using the autocorrelation technique (Levinson-Durbin recursion) and encoded as Line Spectral Frequencies. Each frame is sub-divided into four 40 sample sub-frames (5ms), and, for each sub-frame, the Line Spectral frequencies are interpolated to ensure against discontinuities. The interpolation scheme used is described in Table 3.1.

For every sub-frame an adaptive and fixed codebook search are performed. The adaptive codebook is of length 128 and the fixed codebook contains 1024 overlapped Gaussian codes with an overlap of 2.

For experimentation purposes the gains and LSF parameters are unquantised in the standard coder. However, a suitable bit allocation for the standard Time Domain CELP coder is shown in Table 3.2. The total bits/frame of 142 samples translates to 7.1kbit/s for a frame length of 160

samples. In a practical implementation this bit rate would be increased to some 8kbit/s by the requirements for error correction and synchronisation information.

Parameter	Bits/Sub-frame	Bits/Frame
LSFs: (Federal Standard 1016 quantisation scheme)	_	34
Adaptive Codebook: Gain	5	20
Codebook Index	7	28
Fixed Codebook: Gain	5	20
Codebook Index	10	40
Total Number of Bits:	**27**	**142**

Table 3.2: Suitable Bit allocation for the standard Time Domain CELP Coder.

3.6 Measures for speech coding

The measures used for assessment of speech coders are divided into two classes; Objective and Subjective measures [27]. Objective measures are more easily used since they can be described purely mathematically, while Subjective measures are based on the opinions of a panel of listeners. However, a perfect model of the human ear, and hence a perfect objective measure, does not currently exist. Current objective measures can, thus, only make approximations to the 'real' speech quality perceived by a listener.

3.6.1 Objective speech measures

Three standard objective measures, Average SNR, Segmental SNR and Cepstral Distance, are employed in the work described. A further measure, based on a perceptual auditory model, is derived in chapter 5.

The problem with objective measures is that the human ear does not perceive sounds as a pure set of samples. It is sensitive to particular forms of distortion while other types may not be perceived due to masking and threshold effects. Thus normal communications systems measures must be used with care. The three measures considered were chosen because of their widespread acceptance in the speech coding community.
In general, Objective measures divide the input speech records into frames. All three measures discussed are based on the standard coder frame length of 160 samples, which exploits the fact that speech can be regarded as quasi-periodic over such lengths. Typically, the measures are determined as averages across 20 sentences of mixed male/female speech lasting approximately 1 minute.

Average SNR (AV. SNR)

The Average SNR [28] is the mean value of a large number of frame SNRs. Each frame SNR is a measure of the reconstruction error between the synthesised speech *y(n)* and the input *x(n)*.
The Average SNR is determined as:

$$SNR_{FR}^{m} = \frac{\sum_{n=1}^{M} x^2(n)}{\sum_{n=1}^{M} (x(n) - y(n))^2} \qquad \text{.....(3.44)}$$

The AV. SNR measure is then determined as the mean over the F frames under consideration such that:

$$AV.SNR = 10 \log_{10} \frac{\sum_{m=1}^{F} SNR_{FR}^{m}}{F} \qquad \text{.....(3.45)}$$

Segmental SNR (SEGSNR)

The Segmental SNR [28][29] measure has gained wide acceptance as it compensates, simply, for the under-emphasis of weak signal performance in the AV. SNR measure. The SEGSNR uses dynamic time-weighting, specifically log weighting, converting the SNR values to dB prior to the averaging operation. This ensures that very high SNR values corresponding to well-code high energy speech segments do not camouflage the coder's performance for weak segments.

The SEGSNR measure is defined as:

$$SEGSNR = \frac{1}{F}\sum_{m=1}^{F} 10\log_{10}[SNR_{FR}^{m}] \qquad \text{.....(3.46)}$$

where SNR_{FR}^{m} is the SNR defined by equation (3.44).

While some authors [28] ignore 'silence' frames (frames containing no speech) when calculating the SEGSNR, in this thesis all frames are considered. This gives a more realistic 'feel' for the coder performance in a mobile radio environment where silence frames are unlikely to exist due to background noise.

Cepstral Distance (CD)

The Cepstral Distance [29][30][31][32] differs from the previous measures in that it is a spectral distortion measure. Other definitions exist for the Cepstral Distance (CD), but in this thesis the CD is calculated from the LPC coefficients. This is a convenient and fast technique for comparing the smoothed spectra of the input and output speech signals.

The Cepstral coefficients are derived from the LPC coefficients by relating the Cepstral LPC model to the standard LPC derivation.

The Cepstral LPC model is defined as:

$$\log_e(A(z)) = -\sum_{k=1}^{\infty} c(k) z^{-k} \qquad \text{.....(3.47)}$$

where *A(z)* is the standard LPC filter described previously and *c(k)* are the Cepstral coefficients. The simplest method of calculation for the Cepstral coefficients is a recursion relating them to the standard LPC parameter *a(k)*, such that:

$$c(n) = a(n) + \sum_{k=1}^{n-1} (k/n) c(k) a(n-k) \qquad 1 \le n \qquad \text{.....(3.48)}$$

The Cepstral Distance is then measured, between the Cepstral coefficients of the input *x(n)* and output *y(n)* speech records, as:

$$CD(x,y) = \frac{10}{\log_e 10} \left[(c_x(0) - c_y(0))^2 + 2 \sum_{i=1}^{P} (c_x(i) - c_y(i))^2 \right]^{1/2}$$

.....(3.49)

In this work, the CD is computed using the first P Cepstral coefficients and throughout a predictor order of P=10 is used.

The LPC Cepstral Distance has been shown to correspond well with the subjective Mean Opinion Score (MOS) measure. However, results in this thesis, show that it does not perform well for certain forms of spectral distortion.

3.6.2 Subjective Measures

The most widely used subjective listening test is known as the Mean Opinion Score (MOS) [27]. A large number of listeners are required to rank the synthesised speech on a five point scale:

Rating	Speech Quality	Level of Distortion
5	Excellent	Imperceptible
4	Good	Just perceptible, but not annoying
3	Fair	Perceptible, and slightly annoying
2	Poor	Annoying, but not objectionable
1	Unsatisfactory	Very annoying and objectionable

Two phases of the tests are performed - 'training' during which listeners hear signals representing 'high', 'low', and 'middle' judgement categories. This 'anchors' the opinion scores of the listeners. Then in the second phase -'evaluation' the subjects listen and rank the signal samples according to the above table.

For the tests to be valid it is necessary to have a large set of standard reference signals and to perform substantial training. The problem with the scheme is that different listeners will have different 'goodness' meanings. This will be influenced by the perception of each listener and the types of distortion present in the signals.

Recent work [33] has shown that MOS tests can only achieve meaningful results when performed on a well-trained, experienced listener base. This ensures that the listeners have experience of the types of distortion produced by coders and have a similar 'anchor' for goodness. A typical listener base would exceed 60 people.

Within the scope of this work it was clearly impractical to perform valid subjective testing. Since a number of coders were produced over a 3 year span, the cost and time requirements of such tests would be preventively high. Currently only coders being tested for the mobile cellular and

telecommunications standards (e.g. GSM [18], US Federal Std 1016 [16]) have full subjective tests performed upon them.

3.7 Speech Database

The speech coders described in this work were tested on a speech database constructed by the author. The database consists of some 30 English speakers who were asked to read a list of 20 phonetically balanced sentences taken from the Harvard list [34]. The sentences were then edited into five one minute speech records comprising all 20 sentences. These records contain an equal mix of male and female speakers. Since the records do not use 'Broadcast' speech (e.g. BBC shipping transmissions) they were considered a fair representation of inputs to a mobile telephone. In this thesis the speech records are referenced as the Bath Speech Records 1-5.

The Objective measures, considered in the previous section, were tested on the Standard Time-Domain CELP speech coder, using the Bath Speech Records; results are tabulated in Table 3.3.

	Objective Speech Measure:		
Bath Speech Record:	**AV.SNR (dB)**	**SEGSNR (dB)**	**CD (dB)**
1	12.41	11.62	2.67
2	12.07	11.37	2.66
3	12.06	11.78	2.65
4	12.08	11.53	2.65
5	12.10	11.40	2.66

Table 3.3: Objective Speech Measures for the Standard Time Domain CELP Coder using the Bath Speech Records as input samples..

3.8 Summary

This chapter has described a digital model of speech production. Central to this model is the all-pole synthesis filter, the parameters of which are determined by Linear Predictive analysis. Two Linear Predictive filters were defined; the analysis filter generates the LPC residual from the input speech, and the synthesis filter performs the inverse operation. The coefficients of these filters (the LPC coefficients) are not suitable for direct quantisation and must be transformed to other representations; three alternative forms were described, of which the Line Spectral Frequencies are particularly useful.

While the LPC analysis filter removes the short-term spectral shape of the input speech, a Long Term Predictor was introduced to remove the pitch content. In cascade with the LPC analysis filter, the LTP correlates prior residual sections with the current segment, to maximise prediction gain. The excitation waveform, resulting from the cascade of these filters, can be quantised using Multi-pulse, Regular-pulse or Gaussian sequences. CELP represents the excitation as a vector from a Gaussian codebook, which is chosen using a perceptually weighted MSE search. The perceptual weighting emphasises noise which will not be masked by the speech formants, and hence improves the perceived quality of the synthesised speech.

The Analysis-by-Synthesis CELP architecture can be further improved by use of a closed-loop LTP, known as an Adaptive codebook. This is searched using the perceptually weighted MSE search such that the adaptive code makes an optimal contribution to the output speech.

As a result of the discussions of Linear Predictive Coding, a standard CELP architecture was defined. This searches the codebooks in the Time-Domain and is the basis for the speech coders described in this thesis.

Since human perception is non-linear a simple measure of speech distortion is not available. Three approximate Objective measures were described: AV. SNR and SEGSNR compare the time-domain speech samples, while the CD measures the error between the smoothed spectra. In summary, this chapter has considered the important signal processing used in current speech coders. These signal processing elements are used extensively in the following chapters of this thesis.

3.9 References

[1] L. R. Rabiner and R. W. Schafer, "Digital Processing of Speech Signals," *Prentice-Hall Sig. Proc. Series*, 1978.

[2] J. Makhoul, "Linear Prediction: A Tutorial Review," *Proceedings of the IEEE*, Vol. 63, No. 4, pp. 561-580, April 1975.

[3] J. D. Markel and A.H. Gray, Jr. , "Linear Prediction of Speech," *Communication and Cybernetics 12, Springer Verlag*, 1976.

[4] J. Le Roux and C. Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients," *IEEE Trans. Acoust., Speech, and Signal Proc.*, pp. 257-259, June 1977.

[5] B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *J. Acoust. Soc. Am.*, Vol. 50, pp. 637-655, 1971.

[6] F. Itakura, "Line Spectrum Representation of Linear Predictive Coefficients of Speech Signals," *J. Acoust. Soc. Am.*, Vol. 57, 1975.

[7] B. M. G. Cheetham and P. M. Hughes, "Formant Estimation From LSP Coefficients," *Proc. Fifth Int. Conf. on Digital Process. in Communications, Loughborough, U.K.* , (IERE Publ. No. 82), pp. 183-189, Sept. 1988.

[8] G. S. Kang and L. J. Fransen, "Low-Bit Rate Speech Encoders Based on Line-Spectrum Frequencies (LSFs)," *NRL Report 8857, Naval Research Lab., Washington, D.C.*, Jan. 1985.

[9] P. Kabal and R. P. Ramachandran, "The Computation of Line Spectral Frequencies Using Chebyshev Polynomials," *IEEE Trans. on Acoust., Speech, and Signal Proc.*, Vol. 34, No. 6, pp. 1419-1426, Dec. 1986.

[10] P. Kroon and E. F. Deprettere, "A Class of Analysis-by-Synthesis Predictive Coders for High Quality Speech Coding at Rates Between 4.8 and 16 kbits/s," *IEEE J. on Sel. Areas in Comms.*, Vol. 6, No 2, pp. 334-363, Feb. 1988.

[11] R. P. Ramachandran and P. Kabal, "Pitch Prediction Filters in Speech Coding," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. 37, No. 4, April 1989.

[12] P. Kroon and B. S. Atal, "Pitch Predictors with High Temporal Resolution," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Vol 1, pp. 661-664, 1990.

[13] J. S. Marques, I. M. Trancoso, J. M. Tribolet and L. B. Almeida, "Improved Pitch Prediction With Fractional Delays in CELP Coding," *Proc. IEEE Conf. Acoust., Speech, and Signal Proc.*, Vol 1, pp. 665-668, 1990.

[14] R. E. Crochiere and L. R. Rabiner, "Multirate Digital Signal Processing," *Prentice-Hall Signal Processing Series*, 1983 (Chapters 2 & 4).

[15] J. P. Campbell, T. E. Tremain and V. C. Welch, "The Proposed Federal Standard 1016 4800bps Voice Coder: CELP," *Speech Technology* , pp. 58-64, April/May 1990.

[16] U.S. National Communications System, Washington, D.C. , "Proposed Federal Standard 1016, Second Draft," Nov. 1989.

[17] U.S. National Communications System Washington, D.C., "Details to Assist in Implementations of Federal Standard 1016 CELP," Jan. 1992.

[18] European Telecommunications Standards Institute Technical Committee, "Recommendation 06.10: GSM Full-Rate Speech Transcoding," Version 3.2.0, Jan. 1990.

[19] B. S. Atal and J. R. Remde, "A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates," *Proc. IEEE Conf. Acoust., Speech and Signal Proc.*, pp. 614-617, 1982.

[20] M. R. Schroeder and B. S. Atal, "Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates," *Proc. IEEE Conf. Acoust, Speech and Signal Proc.*, pp. 937-940, 1985.

[21] B. S. Atal and M. S. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. 27, No. 3, pp. 247-254, June 1979.

[22] P. Kroon and B. S. Atal, "Quantization Procedures for the Excitation in CELP Coders," *Proc. IEEE Conf. Acoust., Speech and Signal Proc.*, pp. 1649-1652, 1987.

[23] W. B. Kleijn, D. J. Krasinski and R. H. Ketchum, "Fast Methods for the CELP Speech Coding Algorithm," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. 38, No. 8, pp. 1330-1342, Aug. 1990.

[24] D. Lin, "Speech Coding Using Efficient Pseudo-Stochastic Block Codes," *Proc. IEEE Conf. Acoust., Speech and Signal Proc.*, pp. 1355-1357, 1987.

[25] R. A. Salami, "Binary Code Excited Linear Prediction (BCELP): New Approach to CELP Coding of Speech without Codebooks," *Electronics Letters*, Vol. 25, No. 6, pp. 401-403, March 1989.

[26] C. S. Xydeas, M. A. Ireton and D. K. Baghbadrani, "Theory and Real Time Implementation of a CELP Coder at 4.8 and 6.0 KBits/second

using Ternary Code Excitation," *Proc. Fifth Int. Conf. on Digital Process. in Communications, Loughborough, U.K.* , (IERE Publ. No. 82), pp. 167-174, Sept. 1988.

[27] S. R. Quackenbush, T. P. Barnwell III and M. A. Clements, "Objective Measures of Speech Quality," *Prentice-Hall Signal Proc. Series*, 1988.

[28] N. S. Jayant and P. Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video," *Prentice-Hall Signal Proc. Series*, 1984.

[29] N. Kitawaki, M. Honda and K. Itoh, "Speech-Quality Assessment for Speech-Coding Systems," *IEEE Communications Magazine*, Vol. 22, No. 10, pp. 26-33, Oct. 1984.

[30] N. Kitawaki, K. Itoh, M. Honda and K. Kakehi, "Comparison of Objective Speech Quality Measures for Voiceband Codecs," *Proc. IEEE Conf. Acoust., Speech and Signal Proc.*, pp. 1000-1003, 1982.

[31] N. Kitawaki, H. Nagabuchi and K. Itoh, "Objective Quality Evaluation for Low-Bit-Rate Speech Coding Systems," *IEEE J. Sel. Areas in Comms.* , Vol. 6, No. 2, pp. 242-247, Feb. 1988.

[32] A. H. Gray and J. D. Markel, "Distance Measures for Speech Processing," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. 24, No. 5, Oct. 1976.

[33] I. L. Panzer and A. D. Sharpley, "Comparison of Subjective Testing Methodologies for Speech Quality Evaluation," Proc. *IEEE Workshop on Speech Coding for Telecommunications: Digital Voice for the NIneties*, pp. 93-95, Whistler, B.C., Canada Sept. 1991.

[34] "IEEE Recommended Practice for Speech Quality Measurements," *IEEE Trans. on Audio and Electroacoustics*, pp. 225-246, Sept. 1969.

Chapter 4: The Application of the DFT to CELP Architectures

In the CELP coders described in Chapter 3, the LPC excitation was represented by a gain-adjusted vector selected from a Gaussian codebook. Such codebooks can be stored and searched efficiently using algorithms such as those described by Lin [1] and Kleijn et. al. [2]. In the frequency domain, time domain convolution is transformed identically to vector multiplication and the CELP search procedure can, thus, also be performed in the frequency domain.

In [3], Trancoso and Atal describe the efficient implementation of CELP using transforms such as the DFT and Singular Value Decomposition (SVD). In this chapter, CELP schemes employing DFT domain codebook searches are considered, and two novel techniques, for reducing the size of DFT domain codebooks, are presented [4].

The DFT domain CELP structure also allows analysis of 'pseudo-ideal' excitation sequences for CELP architectures. The features that are revealed by the 'pseudo-ideal' excitation have consequences for future coding structures and, in particular, the spectral content of the excitation is considered.

4.1 Discrete Frequency Domain Searched CELP

A variant of the standard CELP architecture considered in chapter 3 searches the fixed codebook in the DFT domain. The basic architecture of such a scheme is shown in Figure 4.1. In this section the important processes of the DFT domain CELP architecture are considered.

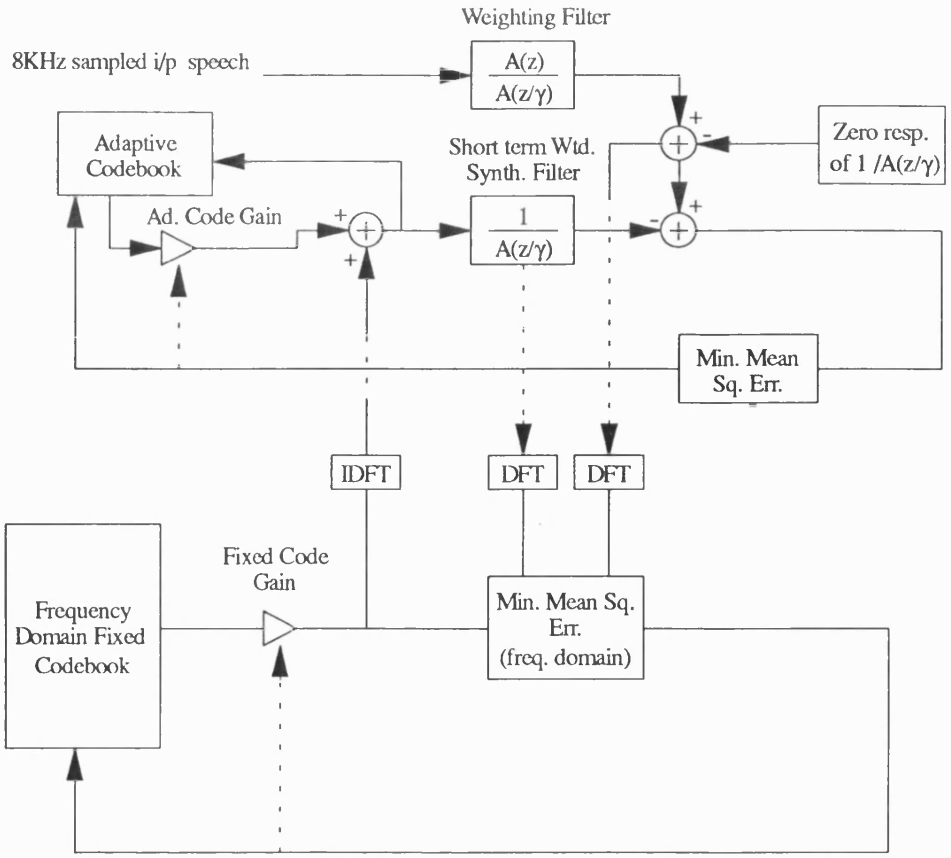


Figure 4.1: Frequency Domain CELP: CELP architecture using a frequency domain fixed codebook search.

4.1.1 Adaptive Codebook Search

Since the adaptive codebook, by its very nature, alters from sub-frame to sub-frame it is impractical to perform adaptive codebook searches in the DFT domain. To do so would require one DFT to be calculated for each code of the adaptive codebook (normally 128) on every search. This is clearly a significant increase in complexity over time domain CELP and would put coder complexity beyond the scope of current DSP processors. (The DSP32C requires 0.099ms to perform a 64 point real FFT: Thus a 128 vector codebook search alone would take approximately 12.7 ms. This contrasts with a sub-frame length of 5ms). Efficient time domain

algorithms for searching the adaptive codebook exist (see section 3.4.5) and these are therefore retained for the DFT domain CELP architecture.

4.1.2 Transformation of the Inverse LPC filter response

In time domain CELP the short term weighted inverse filter used in the codebook search is IIR. For transformation to the DFT domain the impulse response of this filter must be truncated. Fortunately, the impulse response decays rapidly and truncation at 40 samples has negligible effect on the filters behaviour. This can be seen in figure 4.2 which shows the filter response for different speakers and speech segments. The truncation of this filter at 40 samples (5ms) corresponds with the work of Trancoso and Atal [3].

4.1.3 Convolution by DFT domain Multiplication

The time domain convolution is transformed in the DFT domain to a simple vector multiplication, however to maintain equivalence between the domains, circular convolution effects must be avoided. This phenomenon is prevented by zero-padding the time domain sequences prior to the DFT transformation. The required padded length is defined as[5]:

$$L \geq N_1 + N_2 - 1 \qquad \text{.....(4.1)}$$

where N_1 and N_2 are the lengths of the two sequences being convolved. In the case of the CELP search N_1 and N_2 are identically 40 samples, since they are the lengths of the codebook entry $c_q(n)$ and the short term inverse filter response $h(n)$. Thus the minimum value of L is 79 samples. However, for convenience, a transform length (L=$2N$) of 80 samples was chosen. An 80 point DFT can be performed efficiently by using a

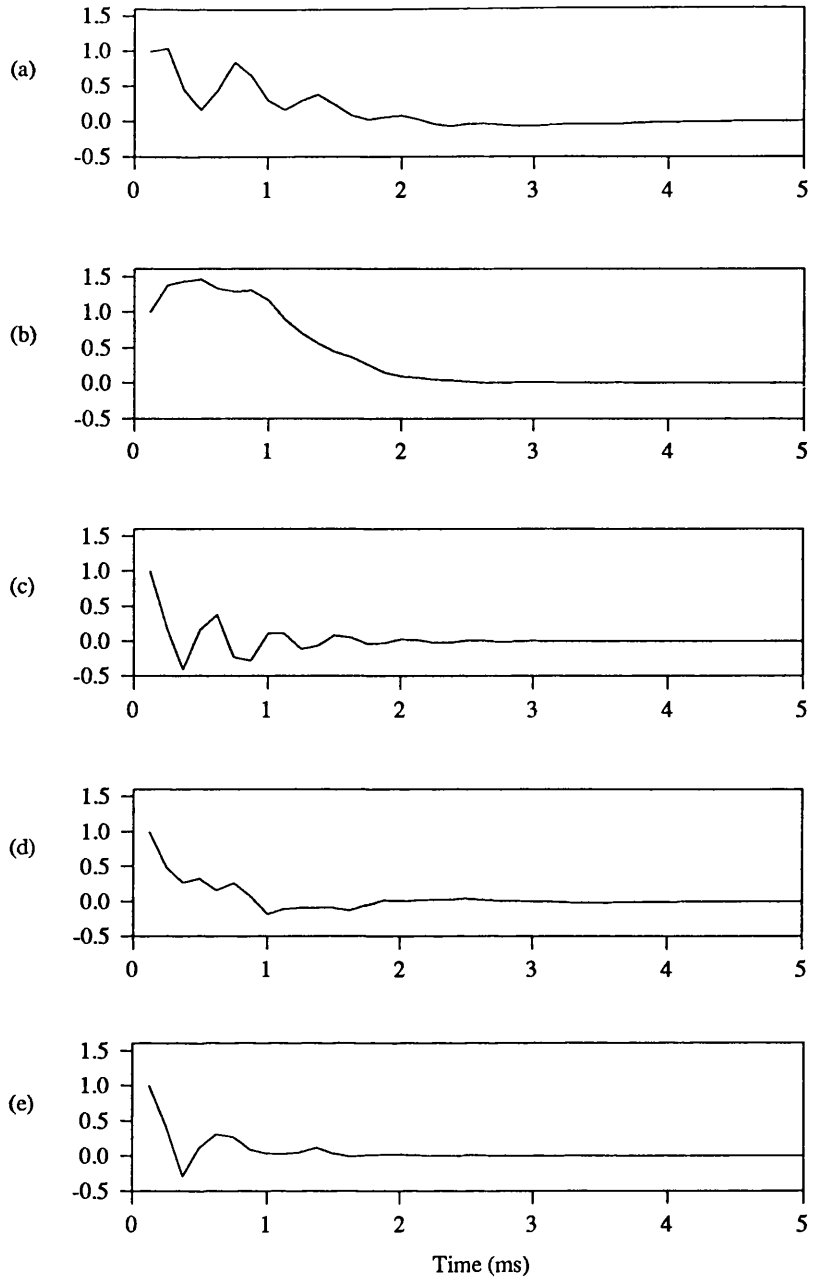


Figure 4.2: Typical Impulse responses of the weighted inverse LPC filter. Note that all responses have decayed substantially within 40 samples (5ms)

composite number FFT algorithm [5] which reduces the DFT to five sixteen point FFTs.

The DFT of a 40 sample sequence, zero padded to 80 samples, consists of the 40 DFT coefficients representing the original sequence interspersed with a second set of 40 interpolated coefficients. This interpolation

process will be considered in section (4.3.2) as the basis of an efficient storage technique for DFT codebooks.

4.1.4 Frequency Domain Codebook Search

The DFT domain codebook search proceeds in a similar manner to the time domain search described in section (3.4.3). The analysis-by-synthesis procedure is best described in matrix-vector notation. In terms of DFTs the total squared error minimisation becomes:

$$E^{(q)} = \sum_{i=0}^{2N-1} \left| \mathbf{X}(i) - \chi^{(q)} \mathbf{H}(i) \mathbf{C}^{(q)}(i) \right|^2 \qquad \text{.....(4.2)}$$

where **X**, **H**, and **C** are, respectively, the DFTs of the weighted input speech, truncated inverse filter response *(h(n))*, and the qth codebook entry under test.

In equation (4.2) the $|\ |^2$ term indicates the norm of the expression such that the equation can be rewritten:

$$E^{(q)} = \sum_{i=0}^{2N-1} \left[\mathbf{X}(i) - \chi^{(q)} \mathbf{H}(i) \mathbf{C}^{(q)}(i) \right] \left[\mathbf{X}^*(i) - \chi^{(q)} \mathbf{H}^*(i) \mathbf{C}^{*(q)}(i) \right]$$

.....(4.3)

where $\mathbf{X}^*$ indicates the conjugate of **X**.

The optimum gain χ is then found, in a similar way to the time domain procedure, by setting the derivative $\delta E^{(q)} / \delta \chi^{(q)} = 0$ in equation (4.3) This leads to an expression for the gain of the form:

$$\chi^{(q)} = \frac{Real \sum_{i=0}^{2N-1} \mathbf{X}^*(i) \mathbf{H}(i) \mathbf{C}^{(q)}(i)}{\sum_{i=0}^{2N-1} \left| \mathbf{H}(i) \mathbf{C}^{(q)}(i) \right|^2} \qquad \text{.....(4.4)}$$

Substitution for $\chi^{(q)}$ in equation (4.3) leads to the search reducing to the minimisation of:

$$E^{(q)} = \sum_{i=0}^{2N-1} |\mathbf{X}(i)|^2 - \frac{\left(Real \sum_{i=0}^{2N-1} \mathbf{X}^*(i)\mathbf{H}(i)\mathbf{C}^{(q)}(i) \right)^2}{\sum_{i=0}^{2N-1} \left| \mathbf{H}(i)\mathbf{C}^{(q)}(i) \right|^2} \quad \text{.....(4.5)}$$

Since the norm of **X** is always positive this expression can be simplified for the purposes of the CELP search to:

$$E'^{(q)} = \frac{\left(Real \sum_{i=0}^{2N-1} \mathbf{X}^*(i)\mathbf{H}(i)\mathbf{C}^{(q)}(i) \right)^2}{\sum_{i=0}^{2N-1} \left| \mathbf{H}(i)\mathbf{C}^{(q)}(i) \right|^2} \quad \text{.....(4.6)}$$

In practice the summations of equation (4.6) can be restricted to N+1 terms by exploiting the conjugate symmetry of the DFTs of real sequences. For calculation purposes equation (4.6) thus becomes:

$$E'^{(q)} = \frac{\left[Real \left(\mathbf{X}^*(0)\mathbf{H}(0)\mathbf{C}^{(q)}(0) + 2. \sum_{i=1}^{N-1} \mathbf{X}^*(i)\mathbf{H}(i)\mathbf{C}^{(q)}(i) + \mathbf{X}^*(N)\mathbf{H}(N)\mathbf{C}^{(q)}(N) \right) \right]^2}{\left| \mathbf{H}(0)\mathbf{C}^{(q)}(0) \right|^2 + 2. \sum_{i=1}^{N-1} \left| \mathbf{H}(i)\mathbf{C}^{(q)}(i) \right|^2 + \left| \mathbf{H}(N)\mathbf{C}^{(q)}(N) \right|^2}$$

.....(4.7)

This expression can be further simplified by removal of the d.c. coefficient, which was found to have little merit in the search process. The Gaussian codebook and the input speech are both generated around a

zero mean, thus making the long term average d.c. component zero. Individual codes may however, have a significant d.c. coefficient value caused by the short code time sequence having a non-zero mean. This was found to sometimes distort the search process; all d.c. coefficients are thus zeroed.

A zero d.c. coefficient in the code vectors leads to a simplified version of equation (4.7):

$$E'^{(q)} = \frac{\left[Real\left(2.\sum_{i=1}^{N-1} \mathbf{X}^*(i)\mathbf{H}(i)\mathbf{C}^{(q)}(i) + \mathbf{X}^*(N)\mathbf{H}(N)\mathbf{C}^{(q)}(N) \right) \right]^2}{2.\sum_{i=1}^{N-1} \left|\mathbf{H}(i)\mathbf{C}^{(q)}(i)\right|^2 + \left|\mathbf{H}(N)\mathbf{C}^{(q)}(N)\right|^2} \quad \text{.....(4.8)}$$

The resulting DFT domain search expression (4.8) is substantially similar to the equivalent time domain expression. However, the searches will not be identical due to the truncation of the IIR inverse filter response. Further, the frequency domain search actually compares the 80 sample convolution result with the zero-extended 40 sample input speech vector. These differences are, however, insignificant, when the short decay of the IIR filter is considered.

4.2 Complexity of the Frequency Domain Search

The complexity of Frequency Domain searched CELP is lower than that of Full codebook searched Time Domain CELP. Comparative measures of complexity are shown in Table 4.1. The measures are based on the numbers of multiplications and divisions required for the search equations (3.37) and (4.8). Each code vector search can be fully described by the evaluation of these expressions. Note that the relevant gain

Process:	Full Search. Time Domain CELP	Overlapped Time Domain CELP	DFT Domain searched CELP	Transformed DFT Domain CELP
Search Overhead	_	882	5280	5280
Code vector Search	882N	242(N-1)	242N	722N
Total (N=1024)	903,168	248,448	253,088	744,608

Table 4.1: Computational Complexity of various codebook search techniques.
(Complexity measured in terms of number of multiplications/divisions).

computations are included within these calculations. The use of multiplication and division operations as a complexity measure is clearly an approximation. The complexity calculations for Frequency Domain searched CELP will probably be artificially low since the manipulation of complex arrays is more processor intensive than that of the real-valued arrays required for Time CELP searches.

In the table the search overhead for Overlapped time domain CELP represents one full convolution evaluation of (3.37) and, for the DFT domain searches, 3 DFT calculations. These 80 point DFTs are computed as five sixteen point FFTs using the 'composite number' technique described in [5]. Since the 80 point DFT becomes the solution of:

$$X(k) = \sum_{l=0}^{4} e^{j\frac{2\pi kl}{80}} \sum_{r=0}^{15} x(5r+l)\, e^{j\frac{2\pi k5r}{80}} \qquad k = 0,1,....79 \ .....(4.9)$$

Equation (4.9) can then be performed as 5 16-point FFTs and 5x80=400 complex multiplications giving a total number of multiplication operations per 80 point DFT as $4x400+5.(16/2)\log_2 16 = 1760$.

The results show that Overlapped Time Domain CELP has the lowest complexity but is very closely followed by DFT domain searched CELP. DFT domain CELP using a time domain codebook, and transforming each vector, is still faster than standard CELP but is impractical for real-time implementations with current processor technology.

In conclusion, the DFT domain search is an efficient alternative to overlapped codebook techniques for CELP searches. It also has the advantage that filtering operations become simple multiplicative weightings of coefficients. The figures for DFT domain complexity could also be significantly reduced by choosing a frame length of a power of 2. This would allow the use of a standard FFT for the transforms and produce a significant reduction in complexity.

4.3 DFT Domain Codebooks

During the frequency domain CELP search it would be possible to individually transform the time domain codebook vectors for each stage of the codebook search (as shown in Table 4.1). This procedure is clearly inefficient, and it is desirable to hold the codebook permanently in the DFT domain. Using a DFT domain codebook, it is necessary only to inverse transform the chosen codebook vector, thus reducing the number of transforms to just one rather than, perhaps, 1024. Some authors [6] have suggested holding both a frequency domain and equivalent time domain codebook. This approach is not considered here since it requires unnecessarily large memory space and offers little advantage over the single transform method.

4.3.1 Full Discrete Frequency Domain Codebooks

For the DFT codebook search, described previously, a Full Frequency Domain codebook would consist of N, 80 coefficient DFTs representing zero extended 40 sample Gaussian sequences. If conjugate symmetry and the zeroed d.c. coefficient are considered, each of the DFT vectors can be described by just 40 complex values. Then, a length N=1024 codebook would require 4x2x1024x40 = 327680 bytes (or 320Kb) of storage space. This calculation is based on each coefficient being stored as two 4-byte IEEE floating point numbers.

For simulation purposes a 320Kb codebook is feasible, however in a mobile telephony environment, where power consumption and physical size are important, such a memory requirement is clearly infeasible. To make DFT searched CELP a practical proposition, it is necessary to derive a reduced size codebook.

4.3.2 Overlapped Frequency Domain Codebooks

So as to avoid circular convolution the transforms in the Full Frequency Domain codebook represent zero extended time domain sequences. This zero extension is effectively a rectangular windowing operation and results in the code's DFT being interpolated by a function of the form:

$$\mathbf{F}(i) = \exp j\left[-\frac{(M-1)i\pi}{N}\right]\frac{\sin\left(\frac{i\pi M}{N}\right)}{\sin\left(\frac{i\pi}{N}\right)} \qquad for\ -\frac{N}{2} \le i < \frac{N}{2} \quad .....(4.10)$$

where M is the length of the original sequence, and N the length of the zero extended sequence (i.e. the transform size), in this case 40 and 80 samples respectively. Two basic constraints can, thus, be placed on a reduced size codebook:

- The generated DFT domain codes must correspond to real time sequences.
- The generated 80 coefficient DFTs must represent transforms of zero-extended 40 sample time domain sequences.

The first of these can be achieved by simply ensuring that the generated DFT codes emulate the conjugate symmetry of the DFTs of real sequences. The second constraint can, however only be approximated in a reduced size codebook. The windowing, representing the zero extension, is transformed in the DFT domain to circular convolution of the original DFT with $\mathbf{F}(i)$ from equation (4.10). Circular convolution, however, requires each code to be stored in its entirety, making any codebook size reduction difficult.

An alternative approach would be to store 40 coefficient codes and interpolate them for each code search. This is, however, clearly inefficient and impractical in a fast codebook search. Further, the codebook size is only reduced to 160Kb by such an approach.

So as to reduce the codebook size, while maintaining the codebook search times possible with a full frequency domain codebook, the codes were overlapped in a similar way to overlapped time domain codebooks (see section 3.4.5). In this case every second coefficient is interpolated and the codes overlap by s coefficients.

The qth code $\mathbf{C}^{(q)}$ is then generated from the codebook $\mathbf{V}$ such that:

$$\begin{aligned} \mathbf{C}^{(q)}(i) &= \mathbf{V}[q*s+i] && 1 \le i < M, s \text{ even} \\ \mathbf{C}^{(q)}(i) &= \mathbf{C}^{(q)}(N-i) && M < i < N \\ \mathbf{C}^{(q)}(0) &= 0 \end{aligned} \qquad \text{.....(4.11)}$$

Thus for an overlap $s=2$ and ignoring the zero d.c. coefficient each sequence commences with the third sample (i.e. the second interpolated value) of the previous DFT code vector.

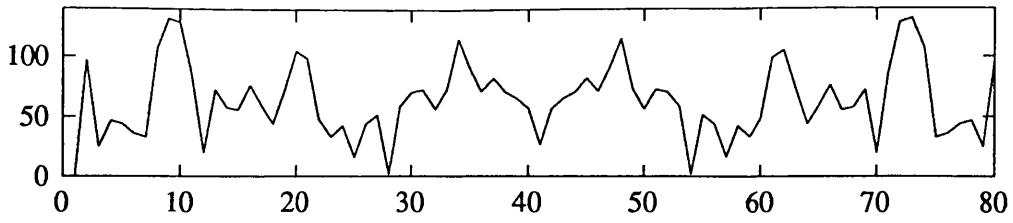
The approximately interpolated coefficients were generated using two techniques:

LONG generates the codebook as a series of DFTs of 512 point Gaussian sequences which are zero extended to 1024 samples to introduce interpolation. In this case, neither the interpolating function nor the circular convolution effects are exactly reproduced. However, the interpolated coefficients are similar to those required.

The second approximation, **CONV**, produces the codebook by interpolating a complex Gaussian sequence with a function of the form of equation (4.10). Here, the function is correct, but the required circular convolution is replaced by linear convolution. In practice, this was found to have little effect on the resulting codes. This is reasonable since, over the code lengths of interest, the Gaussian nature of the sequence will result in linear convolution closely approximating circular convolution.

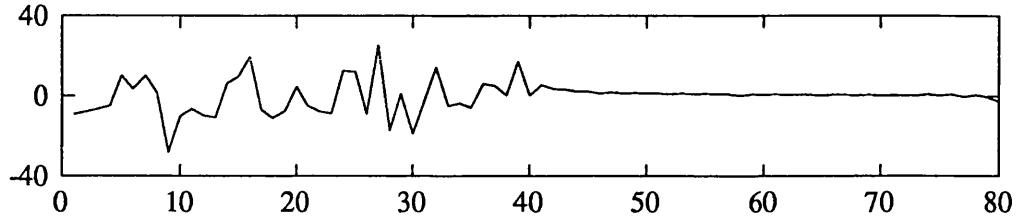
Codebooks generated using either technique require just (1024+40)*8=8512bytes (~8.3Kb) which is approximately 1/40th of the full codebook size. This considerable reduction puts the codebook storage requirements within the capabilities of current DSP processors. Further, it would make the use of Frequency Domain searched CELP practical for mobile telephony.

Both overlapped codebooks were found to generate similar quality codes and examples are shown in Figure 4.3. The approximate nature of the codes can be seen from the equivalent time domain codes shown in the Figure. In particular, the codes are not exactly zero after the 40th sample as would be the case with a full frequency domain coder; the amplitude of these samples is, however, substantially reduced. This is important, since it ensures that the frequency domain codebook search corresponds closely with the time domain convolution approach. Substantial code energy



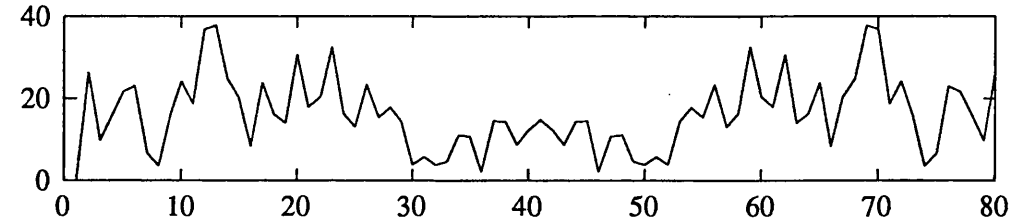
DFT Coefficient

a) Magnitude spectrum of a DFT code derived from codebook CONV.



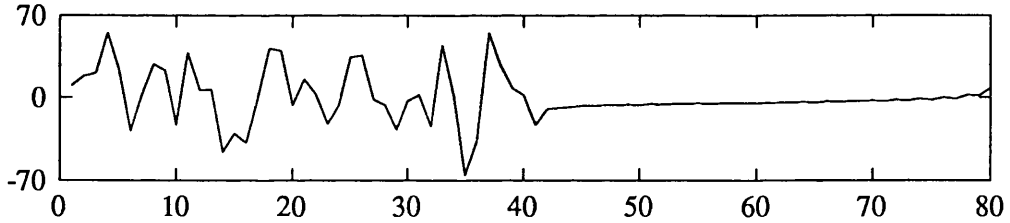
Time Sample

b) Equivalent time domain code derived from codebook CONV.



DFT Coefficient

c) Magnitude spectrum of a DFT code derived from codebook LONG.



Time Sample

d) Equivalent time domain code derived from codebook LONG.

Figure 4.3: Example DFT codes and the time domain equivalents as derived from overlapping DFT codebooks.

beyond the 40th sample would distort the MSE calculation and hence degrade the codebook search process.

In practice, the overlapped frequency domain codebook searched CELP was found to have similar performance to Time domain and Full Frequency domain CELP.

4.4 Results for Overlapped Frequency Domain codebooks

Since frequency domain CELP is equivalent to time domain CELP, the output speech is intended to be synchronous and amplitude matched to the input waveform. This quality makes practical the use of the objective measures (SEGSNR, AV.SNR, CD), discussed in chapter 3.

Codebook Size	Codebook LONG OVERLAP				Codebook CONV OVERLAP				Time CELP	Freq CELP
	2	4	6	8	2	4	6	8		
32	9.64	9.58	9.52	9.61	9.51	9.44	9.50	9.54	9.58	9.56
	9.98	10.03	9.94	9.99	9.79	9.79	9.89	9.90	9.65	9.91
64	10.03	9.98	10.00	9.93	9.79	9.93	9.90	9.93	10.27	10.10
	10.48	10.45	10.44	10.40	10.21	10.42	10.26	10.39	10.39	10.54
128	10.37	10.38	10.35	10.36	10.31	10.31	10.34	10.28	10.55	10.44
	10.93	11.00	10.96	10.90	10.83	10.88	10.90	10.79	10.63	10.98
256	10.73	10.74	10.70	10.71	10.64	10.69	10.63	10.70	11.04	10.80
	11.39	11.33	11.32	11.31	11.31	11.30	11.21	11.34	11.21	11.37
512	11.07	11.04	11.04	11.00	11.01	11.04	11.04	11.05	11.39	11.12
	11.76	11.67	11.76	11.67	11.74	11.79	11.71	11.81	11.71	11.78
1024	11.34	11.35	11.34	11.31	11.38	11.31	11.32	11.32	11.78	11.47
	12.15	12.06	12.01	12.03	12.15	12.01	12.08	12.08	12.06	12.08
	All Results :- SEG.SNR / AV.SNR in (dB)									

Table 4.2: Table of results from Overlapping Frequency Domain Codebooks CONV and LONG for various codebook sizes. All results were generated across 20 Harvard list sentences, spoken by mixed male/female speakers (Bath Speech Record 3).

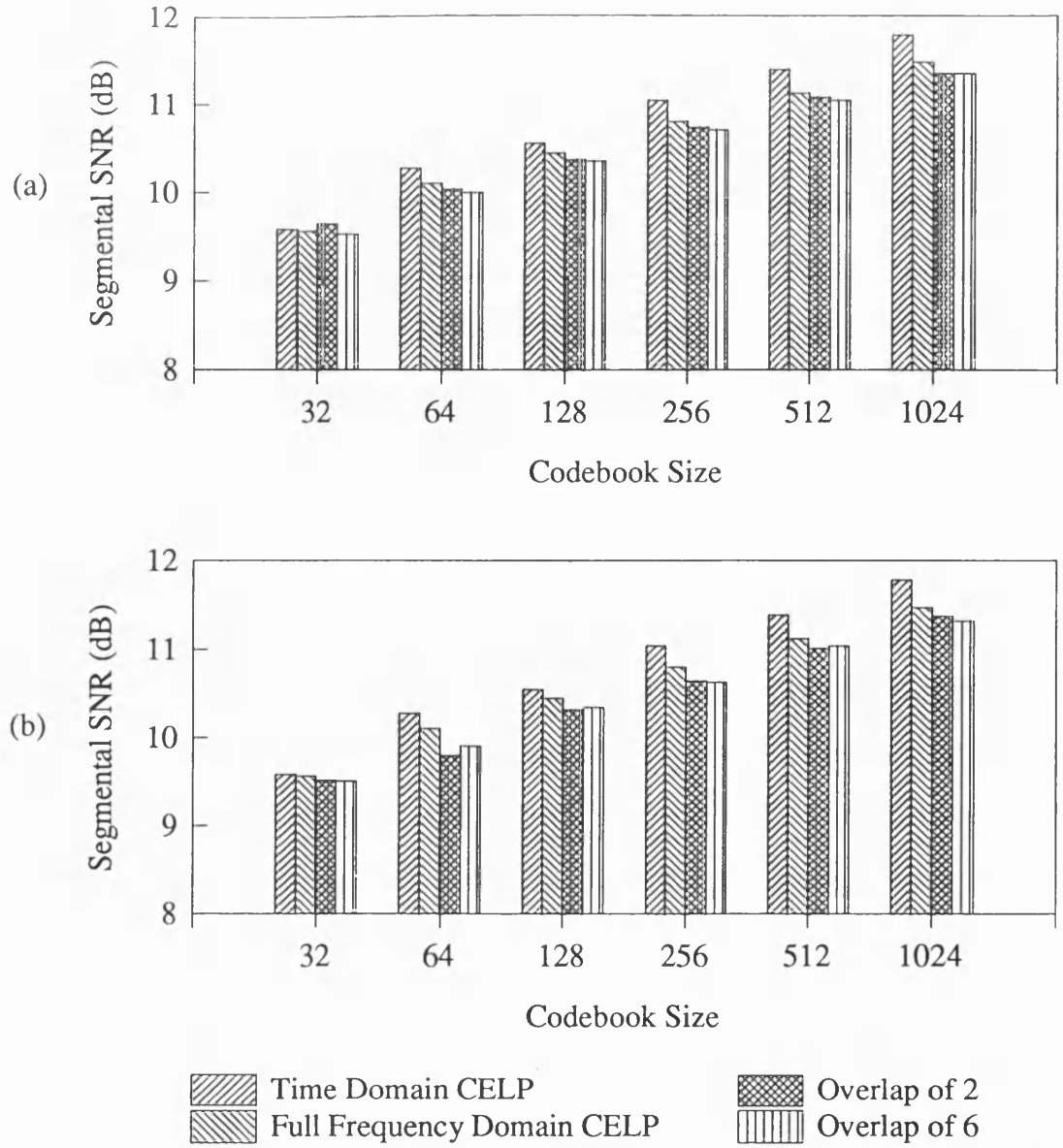


Figure 4.4: Bar Charts showing the results from (a) overlapped codebook LONG and (b) overlapped codebook CONV. For reference results are also shown for Full Frequency and Time domain CELP. All results were generated across 20 Harvard list sentences spoken by mixed male/female speakers (Bath Speech Record 3).

Table 4.2 shows results of various overlapped codebook configurations when the coder is run on the Bath Speech Database record 3. For comparison, results are also included for Time and Full Frequency Domain searched CELP. A selection of results from the table are shown in the bar charts of Figure 4.4. From Table 4.2 and Figure 4.4 it is evident that there is a small degradation in SEGSNR, resulting from the

transformation of the CELP search to the frequency domain. This is caused by the minor differences between the two search processes noted previously. In listening tests the degradation was found to be insignificant.

The results for overlapped codebooks show that SEGSNR results for all levels of overlap, and both codebook types, are close to those of Full Frequency Domain CELP. For a codebook size of 1024 vectors (requiring ~8Kb) the overlapped codebook results are within 0.2 dB of the full 320Kb codebook. At this level of discrepancy, the SEGSNR measure becomes an impractical assessment technique and the only satisfactory method is full listening tests producing Mean Opinion Scores (see section 3.6.2). It was not practical to perform such tests but informal listening tests suggest that there is no audible difference between the performances of overlapped and Full frequency domain codebooks.

Speech waveforms generated using the overlapped DFT codebook architecture are compared, in Figure 4.5, with speech from both full time and frequency domain searches. Again, from the waveforms, it is clear that very little distortion is generated by the overlapped DFT codebook approximations. These results were further confirmed on the other four Bath speech records.

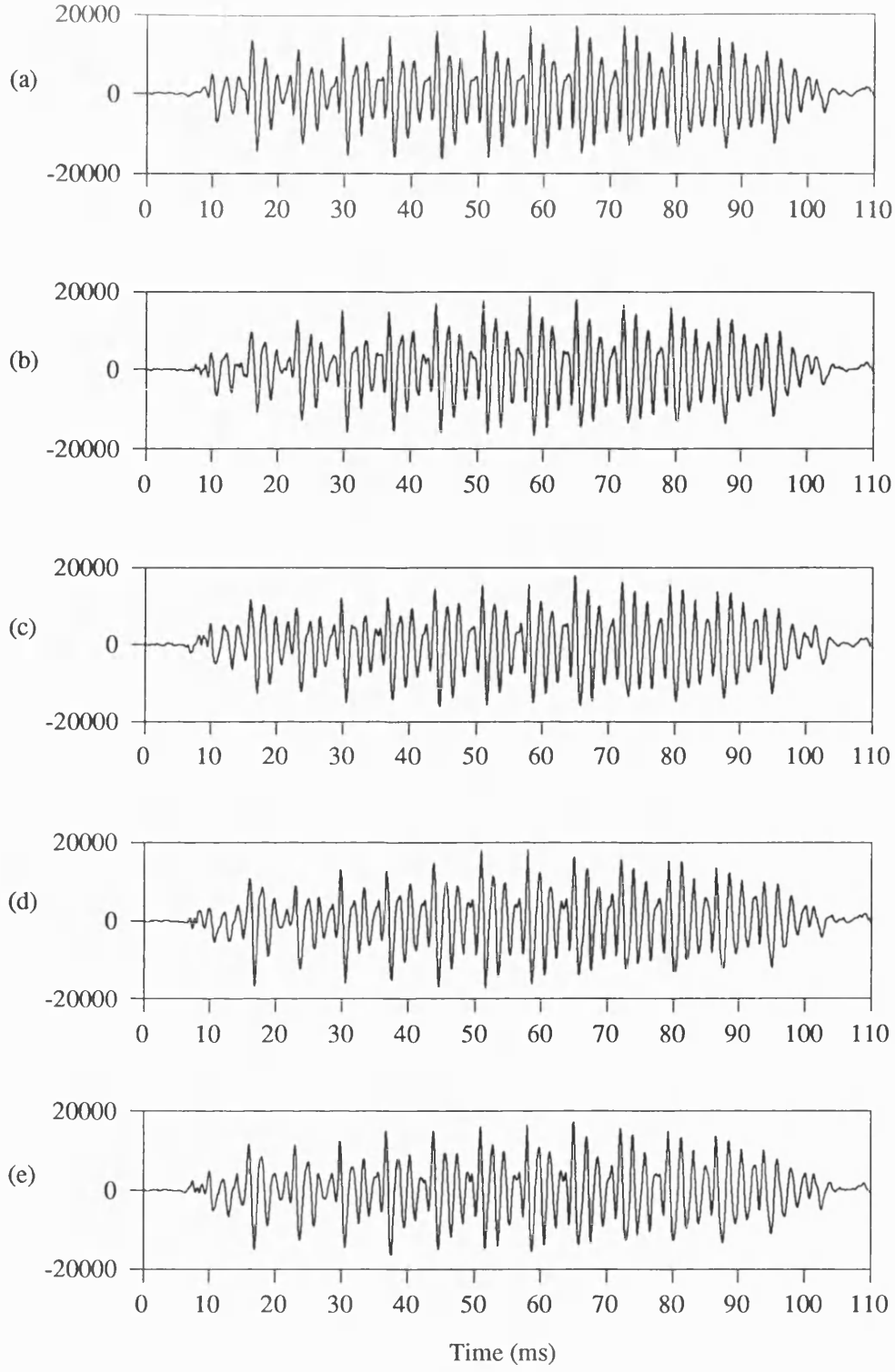


Figure 4.5: Coded speech waveforms from frequency domain CELP using an overlapped DFT codebook: (a) the input speech, (b) output speech using a DFT codebook overlap of 2, (c) output of time domain CELP, (d) output speech using a DFT codebook overlap of 4, (e) output speech using a full frequency domain codebook. The results (c) and (e) are include for comparison purposes.

4.5 DFT Analysis of the LPC Excitation

The Frequency Domain CELP architecture can be adapted to allow calculation of a 'pseudo-ideal' excitation vector for each sub-frame. This vector can be calculated by 'deconvolving' the gain adjusted code from the input speech vector, using the weighted synthesis filter impulse response. In the frequency domain, with some conditions, deconvolution becomes simple division of DFT vectors. Thus, for a given sub-frame, the DFT coefficients of the 'pseudo-ideal' excitation vector $\chi'(k)$are calculated as:

$$\chi'(k) = \frac{\mathbf{X}(k)}{\mathbf{H}'(k)} \qquad for\ 0 \le k < N \qquad .....(4.12)$$

where $\mathbf{X}(k)$ are the DFT coefficients representing the $N=80$ sample zero extended 40 sample input speech vector.

$\mathbf{H}'(k)$ are the DFT coefficients of the $N=80$ sample weighted inverse synthesis filter impulse response.

In time domain CELP, the speech is synthesised as the truncated result of the convolution of the excitation vector and the synthesis filter response. For deconvolution, the convolution result is represented by the 40 samples of input speech, which can only represent a truncated convolution result. This 'ideal' convolution result can be considered as a rectangular windowed 40 sample sequence.

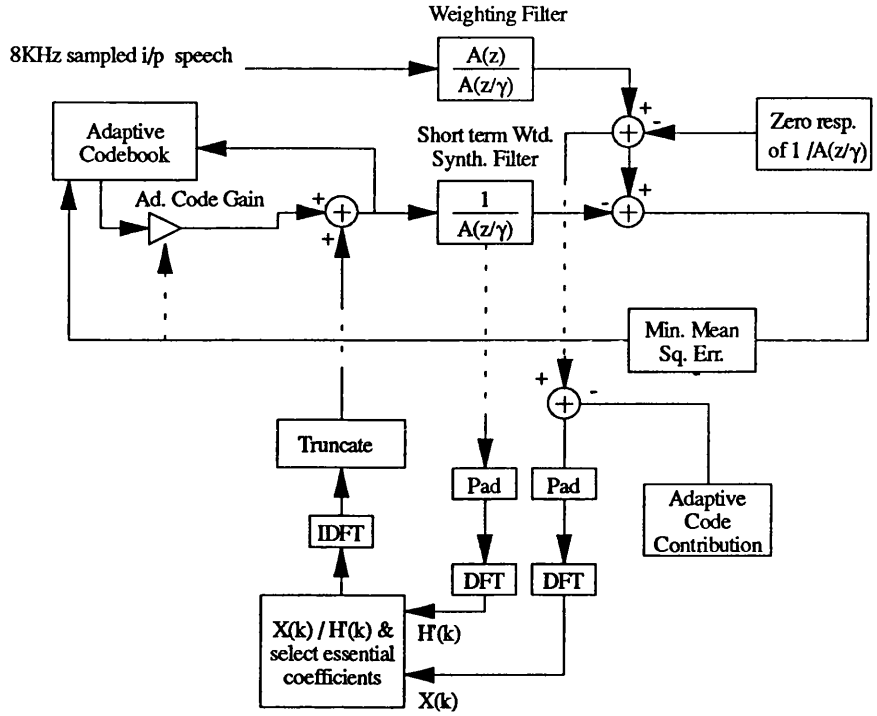
The windowing of the convolution result causes non-cardinal frequencies, present in the input speech, to make non-zero contributions to all $\mathbf{X}(k)$. The time domain 'pseudo-ideal' excitation, $\chi'(n) = \mathbf{IDFT}[\chi'(k)]$, will thus produce a convolution result which is not identical to the input speech sub-frame. The error is characterised by a small variation between the series over the first ~10 samples, though the shape of the speech segment is substantially unaffected. A CELP architecture, using the 'pseudo-ideal'

excitation, produces very high quality output speech, which is indistinguishable from the input.

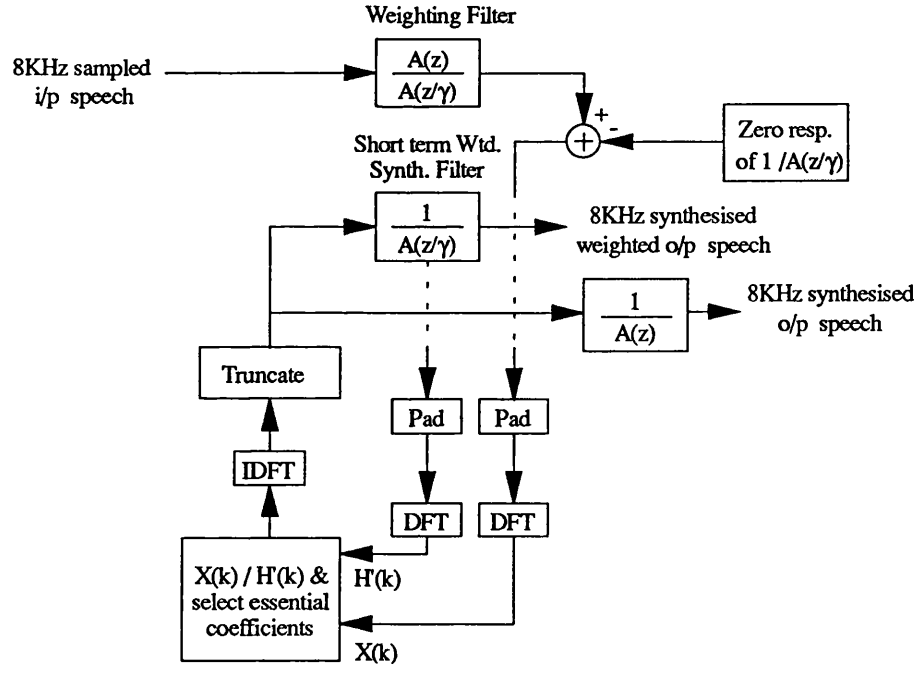
Initially, the 'pseudo-ideal' excitation deconvolution appears to offer a low complexity CELP search technique: A codebook search could be performed on the 'pseudo-ideal excitation avoiding the high complexity convolution operations of standard CELP. However, the inclusion of the error weighting filter in CELP is essential to good code selection and it would also be necessary to include this filter in the 'pseudo-ideal' scheme, This results in the same search complexity as standard Frequency Domain CELP. The 'pseudo-ideal' excitation does, however, allow analysis of the characteristics of the optimum codebook vectors.

The most important 'pseudo-ideal' excitation DFT coefficients will be those corresponding to the maximum magnitude DFT coefficients in the input speech sub-frame. These will be at the input speech spectral peaks and will tend to be at the speech formants. Peaks will also be present among the low frequency coefficients representing the speech pitch content. Since there are only a limited number of significant formants it is interesting to discover how many coefficients of the excitation are significant. The coefficients representing the formants will, however, alter position across the speech record, requiring new coefficients to be selected for each sub-frame.

A limited sub-set of complex coefficients from $\chi'(k)$, corresponding to P spectral peaks of $\mathbf{X}(k)$, were selected. These 'essential' coefficients thus track the positions of the formants in the speech. All other coefficients, including the d.c. component, were set to zero but conjugate symmetry is maintained by including each 'essential' coefficient's conjugate from $\chi'(k)$. The windowing operations cause some unavoidable distortion, but this does not appear to have a significant, audible effect on the synthesised speech.



(a)



(b)

Figure 4.6: Adapted frequency domain CELP architectures allowing 'essential' coefficient analysis. (a) retains the adaptive codebook search for pitch reproduction while (b) depends solely on the 'essential' coefficient excitation..

The frequency domain CELP architecture was then adapted, as shown in Figure 4.6., with the excitation, $\chi'_L(n)$, being derived as IDFT of $\chi'_L(k)$, the 'essential' coefficient version of $\chi'(k)$. No quantisation of gains or coefficients was performed since the aim is to discover the 'information level' required for the excitation.

Two sets of trials were performed, one with and one without an adaptive codebook search. When included, the adaptive codebook contribution is removed from the input speech vector $\mathbf{X}(k)$ prior to the peak analysis. In this case the 'pseudo-ideal' excitation simply replaces a fixed codebook search.

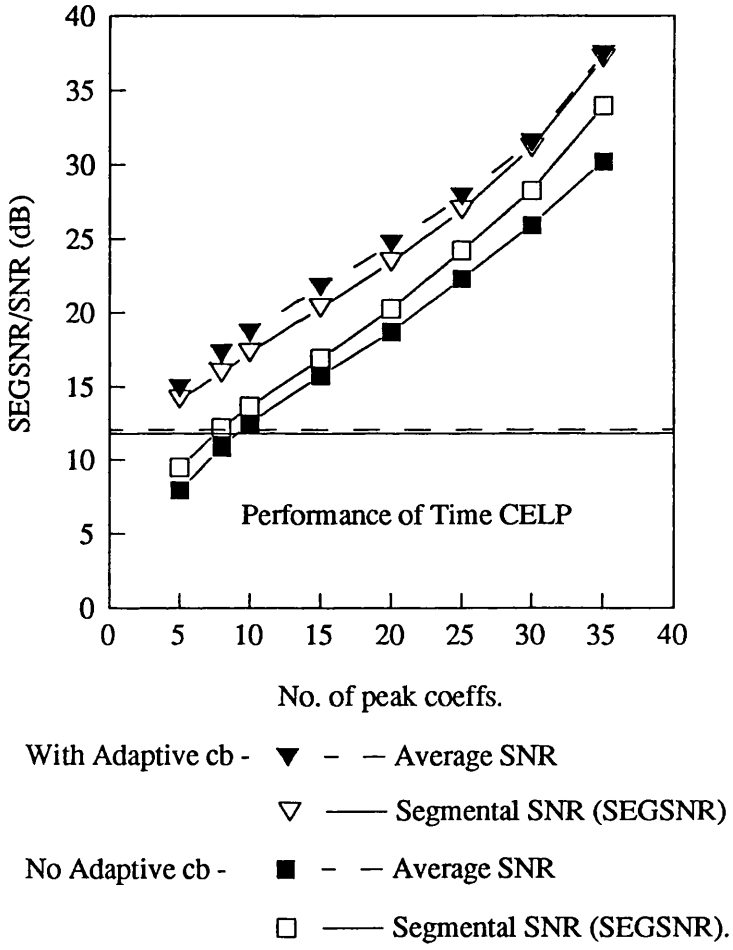


Figure 4.7: Segmental and Average SNRs vs the number of 'essential' DFT coefficients in the innovation sequence. All measures taken across 20 male/female Harvard list sentences (Bath Speech Record 3).

The 'pseudo-ideal' architecture was tested on Bath speech record 3 for 5,8,10,15,20,25,30 and 35 peak coefficients. The output speech quality was measured using both SEGSNR and AV.SNR (see section 3.6.1). These results are shown in the graph of Figure 4.7. A further set of results using the Cepstral distance (CD) are shown in Figure 4.8. These results are of particular interest since the CD measures the spectral distortion of the LPC spectrum.

The graphs of Figure 4.7 shows that, in objective SEGSNR measure terms, just 5 (or ~8 when the adaptive codebook search is excluded) unquantised coefficients are required to reproduce speech of similar quality to CELP. It is interesting to note that Trancoso and Atal [3] also found that five coefficients are required in the SVD domain (Is 5 coefficients a magic number for the excitation ?).

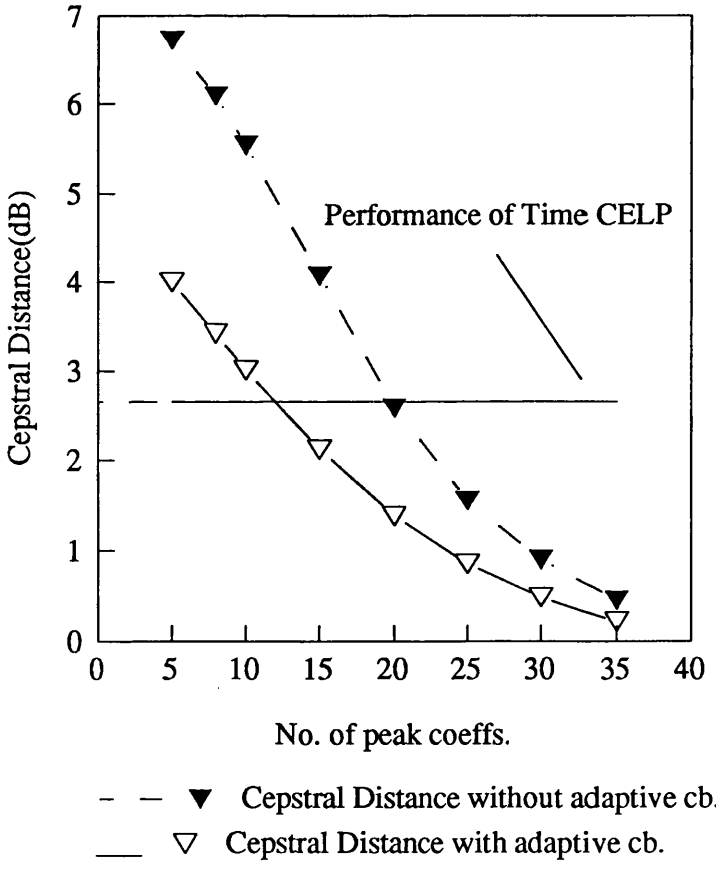


Figure 4.8: Graph showing Cepstral Distance results against the number of 'essential' coefficients retained in the excitation sequence.

The Cepstral distance results of Figure 4.8 are clearly very different from those of the other objective measures. They suggest that, when the adaptive codebook search is excluded, 20+ coefficients are required to equal CELP performance. Listening tests prove that this is not the case and this is clearly a situation where the frequency domain CD measure is an unreliable gauge of coder performance. This is due to the spectral distortion caused by the 'essential' coefficient technique.

It should be remembered that a quantised scheme would require a higher number of coefficients. This makes a low rate quantisation scheme impractical as the position, magnitude and phase of each coefficient would need to be encoded. Such a scheme could be regarded as a frequency domain analogue to multi-pulse coding. The major difference is that multi-pulse schemes have the advantage of coding a set of real values, whilst the 'essential" coefficients are complex. A number of conclusions can, however, be drawn from the result.

Firstly, a 5 coefficient representation of the excitation is substantially a harmonic waveform. Very little of the noisy CELP-type excitation will survive the 'essential' coefficient derivation. Since the 'essential' coefficients track the formants, this suggests that a coding scheme, which concentrates solely on exciting the LPC inverse filter at formants, may be successful. However the coding of coefficient position information is still 'bit' thirsty; the encoding of the positions of just five 'essential' coefficients would require some 25 bits. Such a demand on bits contrasts with the 10 bits required to code a CELP codebook entry.

The second conclusion which may be drawn relates to CELP coding schemes. The premise of CELP is that the excitation is best represented by a Gaussian source combined with a Long Term Predictor to introduce the pitch periodicity. However, the 'essential' coefficient scheme produces high quality speech without the need for either a pitch generator or a

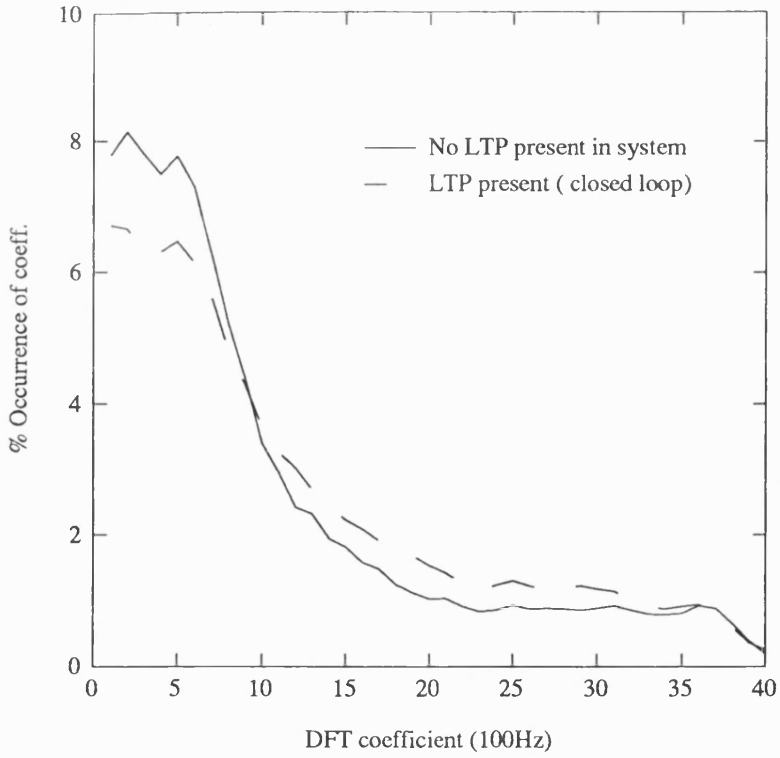


Figure 4.9: Distribution of the DFT coefficients chosen as 'essential' coefficients by measuring the positions of the top ten spectral peaks in the input weighted speech.

Gaussian noise source. Thus the 'essential' coefficients must fulfil both the roles of fixed and adaptive codebooks. So as to investigate the role of the 'essential' coefficients further, the distribution of a scheme using ten coefficients was considered. Over Bath Speech Record 3 the distribution of the positions of chosen essential coefficients was then monitored, producing the results of Figure 4.9.

The distribution shown in Figure 4.9 is significantly 'low pass' even with an adaptive codebook search (LTP) present in the coder. The two configurations produce closely related distributions which serve to confirm the overall results. Since the first five DFT coefficients will represent frequencies of pitch information, the LTP 'present' distribution suggests that there is a substantial pitch component which remains unrepresented by the adaptive codebook search.

Beyond the first five coefficients, the distribution tails off gradually to a near-constant level between the 20th and 35th coefficients. The final five coefficients do not figure significantly in the spectral peak distribution. The second peak in the distribution probably corresponds with the first formant, showing its importance in the excitation. Other formants are, clearly, less important.

The skew of the distribution suggested that there may be a significant relationship between the coefficients chosen in each sub-frame. A further experiment was thus performed whereby a single set of coefficient positions (those representing the first sub-frame in a frame) were chosen as representing all frames. The magnitude and phase components for this set of coefficients were then computed for each sub-frame.

A coder using such a scheme for ten coefficients produced SEGSNR and AV.SNR measures of 14.08dB and 14.58dB, respectively. Informal listening tests confirm these results and suggest that they are, perhaps, low. It is likely that the time domain measures are distorted by the slight slurring which is heard. This is due to the failure to precisely track the formants. While such a scheme would not make a practical coder, the results show that the CELP architecture still leaves significant redundancy in the excitation. Future coding schemes will exploit this to further reduce CELP bit rates.

4.6 Conclusions

This chapter has considered an alternative CELP architecture, which searches a fixed DFT codebook in the DFT domain. Such a scheme, transforms the convolutions of Time Domain CELP to multiplications of DFT vectors. However, the zero-padding, required to avoid circular

convolution in the Time Domain, introduces interpolation of the DFT of each vector.

A full DFT domain codebook requires 320Kbytes of fixed memory, which is impractical for most mobile/portable environments. Two approximate, overlapped DFT codebooks were, thus, introduced which simulate the interpolation effects caused by zero-padding. These new techniques reduce the DFT codebook size to just 8.3Kbytes and were shown to produce no significant degradation when compared to Full DFT domain CELP. The new overlapped codebooks make DFT domain CELP equivalent to overlapped codebook, Time Domain CELP in terms of both computational complexity and fixed memory requirements.

A consequence of the DFT domain CELP architecture is that a 'pseudo-ideal' excitation can be deconvolved from the input speech vector. While this deconvolution offers no computational advantages for the CELP search, it does allow analysis of the characteristics desirable in the CELP excitation. In particular, it was shown that an excitation, consisting of just five 'essential' DFT coefficients, can produce synthesised speech of a similar quality to standard CELP. Even when a pitch predictor was included in the CELP search, the distribution of the 'essential' coefficients was shown to be significantly low-pass and there is, thus, significant inter-sub-frame redundancy. From these results, three conclusions, concerning CELP architectures, can be drawn:

- The adaptive codebook search, while producing adequate speech quality, represents a relatively low degree of the pitch and fundamental frequency information present in speech.
- A simple excitation represented by limited, but perceptually significant spectral information can produce high-quality synthesised speech.

- Standard CELP coders fail to recognise and, hence, exploit the inter-sub-frame redundancies exposed by DFT analysis of the excitation.

In future chapters of this thesis these conclusions are used to derive new, improved coder architectures.

4.7 References

[1] D. Lin, "New Approaches to Stochastic Coding of Speech Sources at Very Low Bit Rates," *in Signal Processing III: Theories and Applications, Elsevier*, 1986.

[2] W. B. Kleijn, D. J. Krasinski and R. H. Ketchum, "Fast Methods for the CELP Speech Coding Algorithm," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. 38, No. 8, pp. 1330-1342, Aug. 1990.

[3] I. M. Trancoso and B. S. Atal, "Efficient Search Procedures for Selecting the Optimum Innovation in Stochastic Coders," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. 38, No. 3, pp. 385-396, March 1990.

[4] I. S. Burnett and R. J. Holbeche, "The Application of the DFT to CELP Architectures," Proc. *IEEE Workshop on Speech Coding for Telecommunications: Digital Voice for the Nineties*, pp. 83-84, Whistler, B.C., Canada, Sept. 1991.

[5] A. V. Oppenheim and R. W. Schafer, "Digital Signal Processing," *Prentice-Hall International Inc.*, 1975.

[6] J. L. Lee and C. K. Un, "On Reducing Computational Complexity of Codebook Search in CELP Coding," *IEEE Trans. on Communications*, Vol. 38, No. 11, pp. 1935-1937, Nov. 1990.

Chapter 5: Analysis by Synthesis Coding with Improved Perceptual Search

Standard Time Domain CELP uses a simple distortion measure, which is conveniently incorporated into the search as a weighted Mean Squared Error computation. The error weighting filter exploits the noise masking properties of the ear, by reducing noise in bands away from the formants. While this simple masking property is important, there are many other phenomena involved in the perception of speech. In the 1930s and 40s, many psychoacoustic experiments were performed [1], and these identified a number of key perceptual processes. These results have been confirmed in more recent, physiological, experiments [2].

This chapter considers analysis by synthesis coders, operating in both the time and frequency domain, which include a distortion measure based on the psycho-acoustic effects displayed by the ear. The aim of the work is to improve the performance of CELP speech coders while not substantially altering the architecture.

5.1 A psychoacoustic perceptual measure

The perceptual measure described in this section is similar to the Bark Spectral Distortion (BSD) described by Wang and Gersho [3][4]. In the latter, the BSD was derived as a spectral, objective, measure for speech which would approximate subjective Mean Opinion Scores (MOS). An objective technique is desirable as MOS measures require substantial numbers of trained listeners, making them impractical for most speech research.

For incorporation into the analysis by synthesis architecture a number of modifications are made to the BSD and further improvements are investigated. The latter exploit the results of [2] and make the measure map the physiology of the ear more closely. Before discussing the incorporation of the BSD into the CELP architectures, the BSD is discussed in some detail.

The BSD emulates several of the known auditory processing features of the human ear, namely:

- Frequency scale warping - performed by a Bark transformation.
- Unequal sensitivity of the ear to different frequencies.
- Non-linear subjective loudness of frequencies.

These perceptual effects are simulated by mathematical processes which lead to a description of a speech frame by a perceptually meaningful parameter vector. Two vectors representing an input and prospective coded frame can then be compared in a perceptual parameter space. This approach is different from most previous techniques in that the measure attempts to recreate the behaviour of the auditory nerve for each speech vector. Since only physically significant vectors are compared, the measure should match the auditory processes more realistically than previous objective measures [5].

The main processes of the BSD are shown in Figure 5.1. The first process is a standard DFT which is used to generate the power spectrum of the input frame $|\mathbf{X}(f)|^2$. All further processes of the BSD calculation are performed on this power spectrum or derivations of it.

While the phase of the speech is ignored by the power spectrum computation, drastic phase changes accompanied by magnitude spectrum changes will be detected. This is a phenomenon common to low-rate speech coders [4]. Experimental results [1] suggest that phase changes alter both the timbre and pitch clarity of sounds, so it is important that

large phase changes do affect the measure. The lack of a complete phase analysis is, however, a possible area of weakness in the BSD measure.

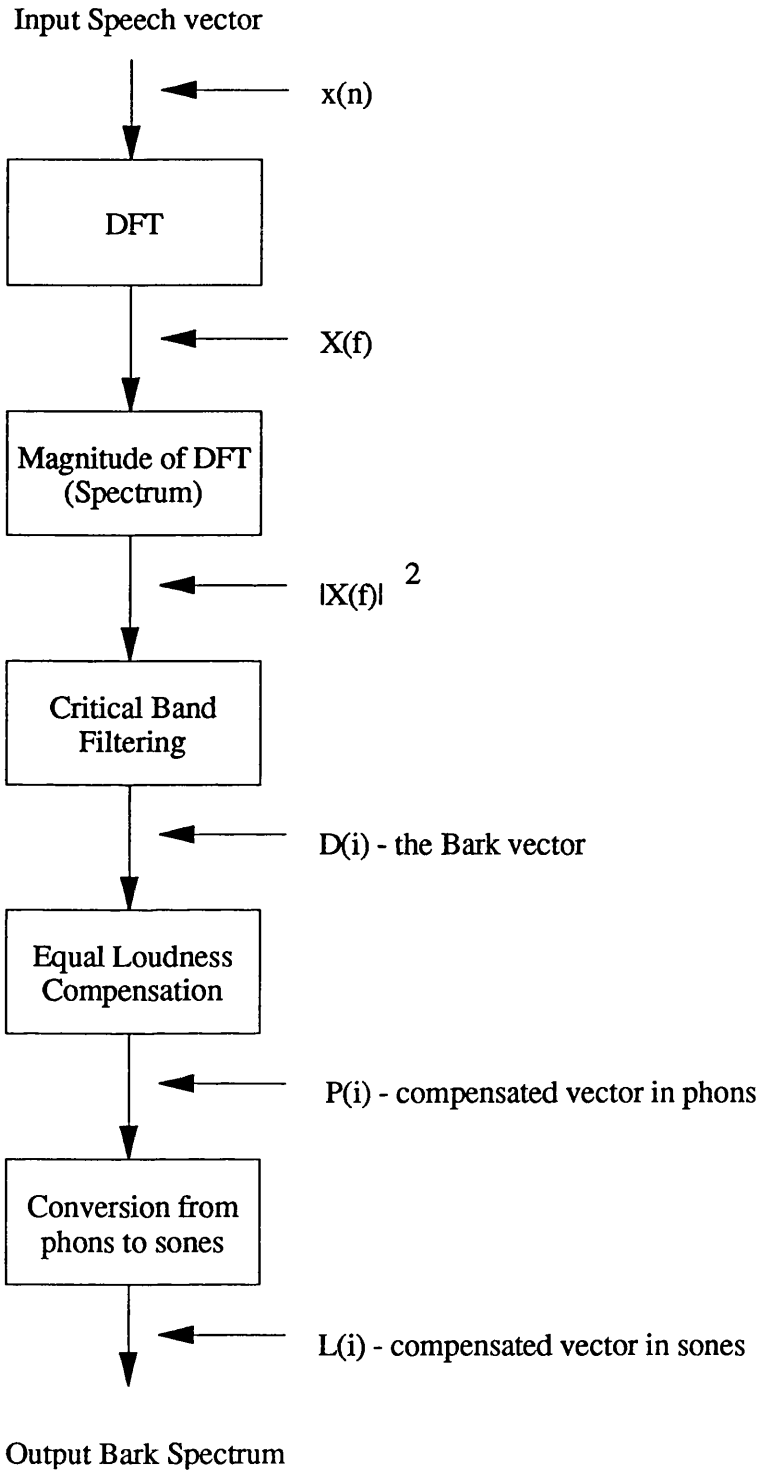


Figure 5.1: Description of processing blocks for Bark Spectrum computation.

5.1.1 Critical Band Filtering

The concept of Critical Bands was introduced by Fletcher [1]. The Critical Band model was detailed in the discussions on masking in section (2.2.6) of this thesis. Simply, critical bands form a model of the perception of sounds, when in the presence of interfering sound sources. Recent work [2] has confirmed the critical band concept with physiological measurements of cat auditory nerves. These show that the tuning curves associated with auditory neurons are substantially similar to the psycho-acoustically measured Critical Bands.

It was clearly impractical to perform physiological experiments on the human auditory nerve. Thus, the filters used here were derived from the psychoacoustic experiments of Zwicker [6][7] whose work was developed by Sekey and Hanson [8] to produce the critical band filter function:-

$$10\log_{10}F(b) = 7 - 7.5(b-0.215) - 17.5\left[0.196+(b-0.215)^2\right]^{1/2} \quad \text{.....(5.1)}$$

This filter function is defined on the Bark scale such that all filter bandwidths are 1 Bark and the filters are then, initially, spaced at 1 Bark intervals [6]. The Bark scale is related to frequency by the non-linear function:

$$f = 600\sinh(b/6) \quad \text{.....(5.2)}$$

Using this transformation the frequency response of a typical Critical Band filter function, centred at 1kHz can be derived, as shown in Figure 5.2.

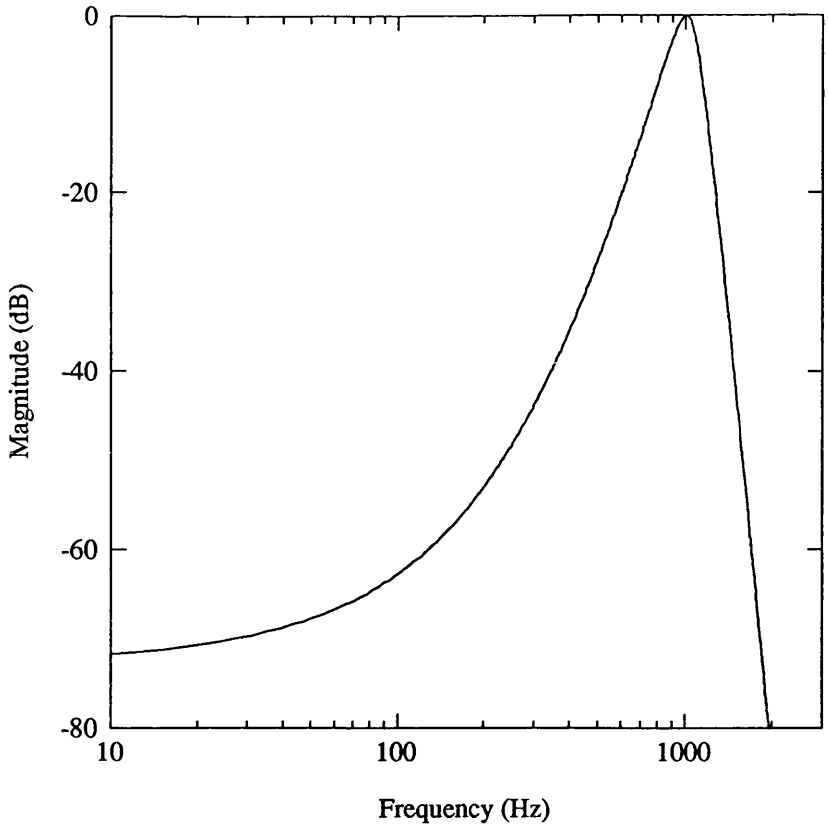


Figure 5.2: A critical band filter function centred on 1kHz.

One problem with the relationship of equation (5.2) is encountered when a bank of 1 Bark spaced filters is constructed - the filters do not cover the required bandwidth efficiently. For convenience, the relationship is thus modified such that:

$$f = 600\sinh((b+0.5)/6) \qquad \text{.....(5.3)}$$

This is permissible since the precise positioning of the critical band filters is unimportant [8]. The rate of addition of the filters is the essential characteristic of the critical band filter bank and this simple modification ensures that all frequencies within the telephone bandwidth are within the 3dB bandwidth of one of the critical band filters.

The centre frequencies of the 1 Bark spaced critical band filter bank are detailed in table 5.1 and Figure 5.3 shows the filter bank on the logarithmic frequency axis.

Filter Number	Centre Freq.(Hz)	Bandwidth (Hz)
1	151.57	102.93
2	257.30	108.37
3	370.19	16.82
4	493.39	128.53
5	630.33	143.82
6	784.81	163.11
7	961.15	186.94
8	1164.25	215.98
9	1399.76	251.02
10	1674.25	293.06
11	1995.35	343.26
12	2372.00	403.02
13	2814.70	473.99
14	3335.77	558.17
15	3949.70	657.88

Table 5.1: The Centre Frequencies and Bandwidths of the Critical Band Filters.

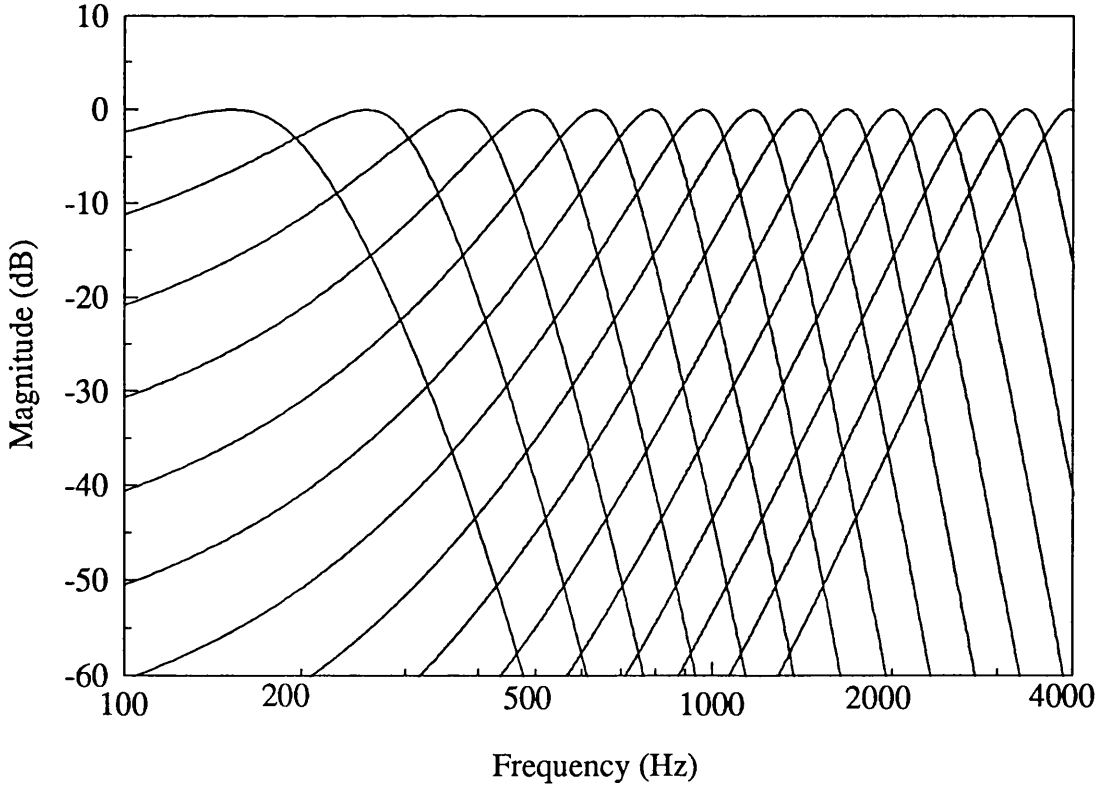


Figure 5.3: One Bark spaced critical band filter bank.

There are thus fifteen filters spanning the 0 to 4kHz bandwidth. While the filter functions could be converted to the linear frequency domain, it is more convenient to perform the Critical Band filtering in the Bark domain. Since the critical band filters are uniform in the Bark domain the process can then be considered as a shifting process, whereby each filter output is generated by shifting the 1 Bark filter through the Bark domain speech spectrum. Using this idea, the filtering operation can be considered as a convolution in the Bark domain such that:

$$\mathbf{D}(i) = \mathbf{F}(i) * \mathbf{Y}(i) \qquad for\ i = 1,2,....,N \qquad .....(5.4)$$

where N is the number of critical band filters (15 for 1 Bark spacing), and **F**(i) and **Y**(i) are the Bark domain filter and speech spectra, respectively.

Bladon [9] describes **D***(i)* as the 'excitation pattern' resulting from the auditory nerve. This result is similar to the excitation computed in [2] by FIR filtering through the measured cats auditory nerve response.

In practice, the calculation of 5.4 is performed by translating the required Bark points back to the linear frequency domain. Hence:

$$\mathbf{Y}(i) = \mathbf{X}(600\sinh((i+0.5)/6))) \quad for\ i = 1,2,.....,N \quad .....(5.5)$$

where **X**() is the previously derived speech power spectrum.

The required points can be predetermined so as to speed up this calculation. As suggested by Hermansky [10], each filter is calculated within a Bark range of -2.5 to 1.3 Barks.

Following the filtering operation the spectrum of the input speech has been smoothed by the critical band functions, resulting in a downsampled parameter vector. However, it is still necessary to include some other non-linearities of the auditory process.

5.1.2 Perceptual Weighting of the ear.

It is well known that the ear does not perceive all tones of a given actual intensity to be equally loud. For example a 100Hz tone may need to be 35dB more intense than a 1kHz tone to be perceived as being equally loud. Complete curves of equal loudness were determined in [11] and are shown in Figure 5.4.

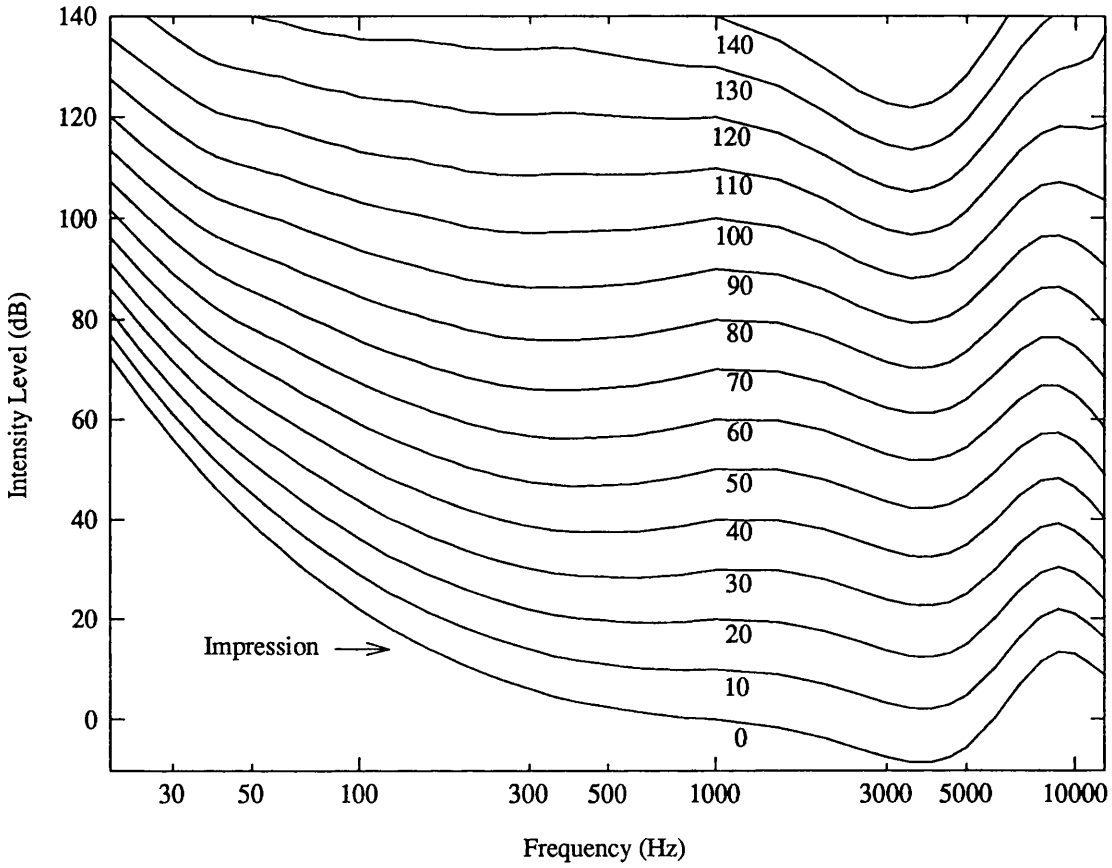


Figure 5.4: Curves of equal loudness for the human ear (after [11])

Each curve, of Figure 5.4, shows how the perceived intensity varies with frequency for given sound intensities, measured as Sound Pressure Level (SPL). The curves allow the 'phon' to be defined as the intensity of a 1kHz tone which would be perceived as being equally loud to a given tone.

It is thus necessary to alter the Bark vector to account for the equal loudness curves. This is performed over the region of interest (telephone speech @ 300-3400Hz, 40-80dB) by a bilinear pre-emphasis filter defined by:

$$\mathbf{H}(z) = (2.6 + z^{-1})/(1.6 + z^{-1}) \qquad \text{.....(5.6)}$$

This operation is performed in the Bark domain, by calculating values for this filter at the Bark domain filter positions. This reduces the complexity of this process and contrasts with Wangs approach [4]. Hence the compensated Bark vector $\mathbf{P}(i)$, in Phons, is derived as:

$$\mathbf{P}(i) = \mathbf{D}(i)\mathbf{H}(i) \qquad for\ i = 1,2,....,N \qquad \text{.....(5.7)}$$

5.1.3 Subjective Loudness

The final adjustment of the process is for subjective loudness. The human ear does not perceive quiet tones and loud tones linearly. The Bark vector, $\mathbf{P}(i)$, is thus converted from Phons to Sones. The Sone scale is a true perceptual scale of loudness, which is divided into two sections around a threshold of 40dB. The definition of a Sone is 'a doubling of perceptual loudness' and the conversion is performed by the curve of Figure 5.5 and the following equation:

$$L = \begin{cases} 2^{(P-40)/10} & for\ P \geq 40 \\ (P/40)^{2.642} & for\ P < 40 \end{cases} \qquad \text{.....(5.8)}$$

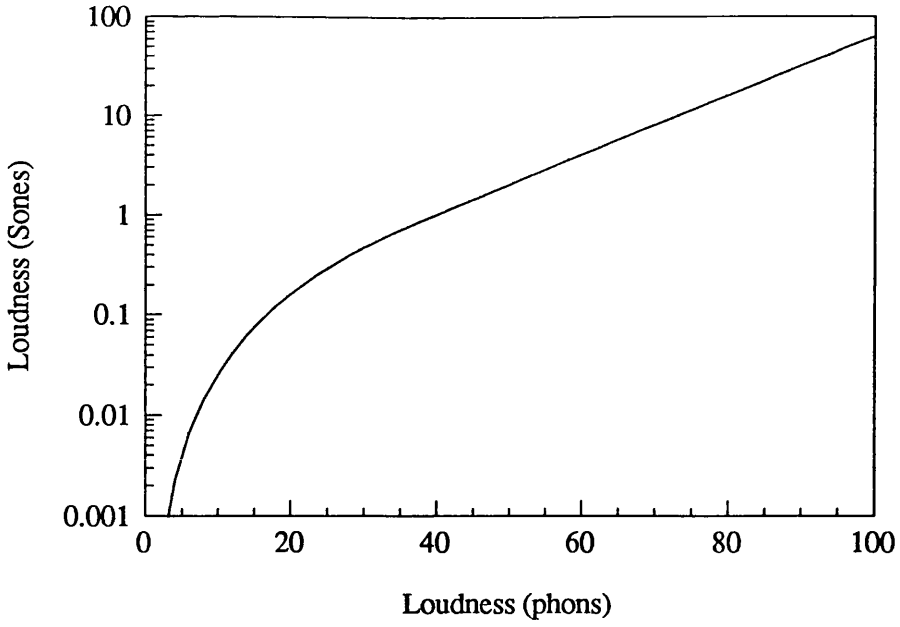


Figure 5.5: Psychophysical scale of loudness - conversion of phons to sones.

Unfortunately, this equation requires a knowledge of the 40dB point for the current speech record when played to the listener. Clearly this is impractical, and an assumption is necessary. Wang [4] states that telephone speech averages 78dB and seldom falls 35dB below this average. Thus the second part of equation 5.8, corresponding to the upper section of the curve of Figure 5.5 is used exclusively. The final processed Bark vector, $\mathbf{L}(i)$,now becomes:

$$\mathbf{L}(i) = 2^{(\mathbf{P}(i)-40)/10} \quad for\ i = 1,2,.....,N \qquad .....(5.9)$$

5.1.4 Bark Spectral Distortion Measure

Following the adjustments and calculations of the previous sections the critical band filtering process results in a vector, $\mathbf{L}(i)$, in Sones, representing the output of each of the Critical Bands. This vector can now be compared with a second vector so as to produce a measure of the spectral distortion (or distance) between the vectors.

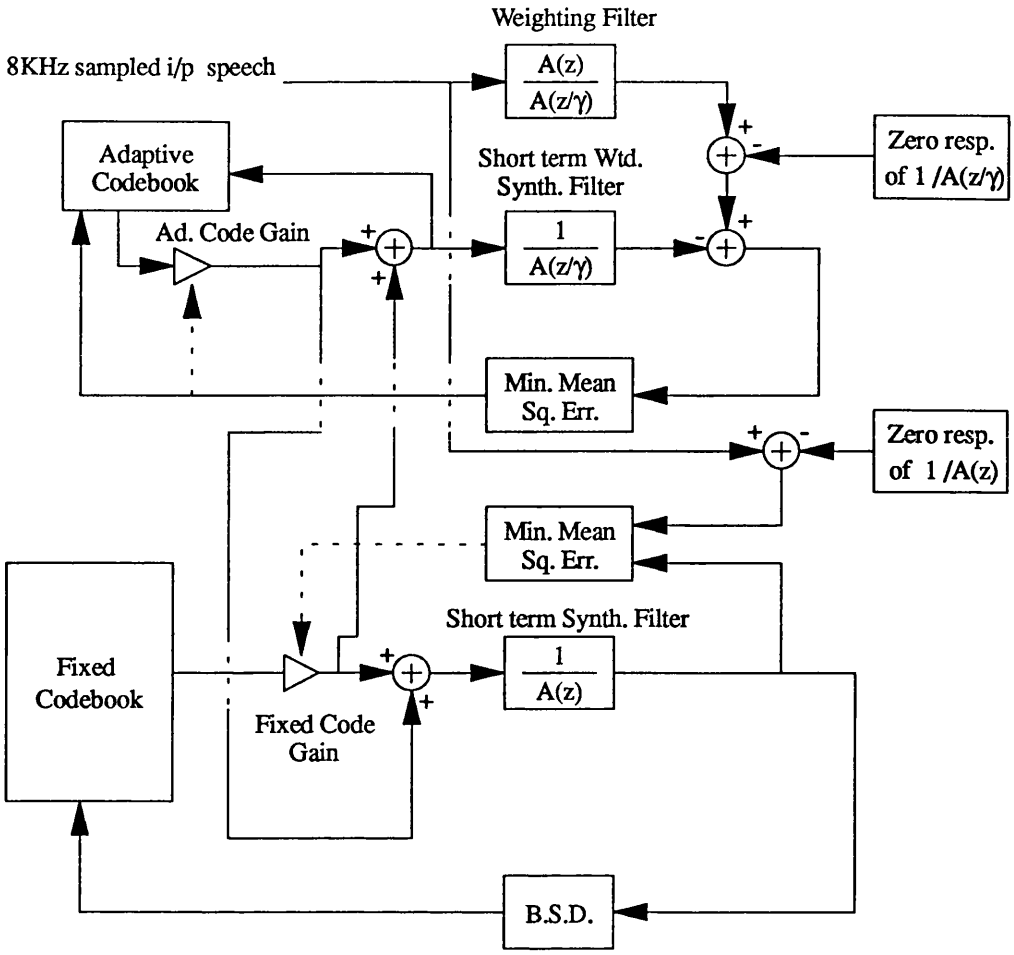


Figure 5.6: The Time Domain CELP architecture adapted to use a mixed BSD/MSE search. The BSD is used for the fixed codebook search and the MSE retained for the adaptive codebook.

The Bark Spectral Distortion measure is defined as:

$$\mathbf{BSD}^{(k)} = \sum_{i=1}^{N} \left[\mathbf{L}_x^{(k)}(i) - \mathbf{L}_y^{(k)}(i) \right]^2 \qquad \text{.....(5.10)}$$

where

N - the number of Critical Band filters.

$\mathbf{L}_x^{(k)}(i)$ - Bark spectrum of the kth segment of the input speech

$\mathbf{L}_y^{(k)}(i)$ - Bark spectrum of the kth segment of the prototype speech

The BSD will thus be minimised for vectors which are close, when compared in the Bark domain.

5.2 The Incorporation of the BSD into Time Domain CELP

The incorporation of the BSD into the standard Time Domain CELP architecture discussed in chapter 3 is, superficially, simply the direct replacement of the MSE calculation with the BSD. However, requirements placed upon the BSD computation lead to a number of significant changes to the codebook search process.

Throughout the following discussion, the adaptive codebook search is retained and performed using the standard MSE search process. A sub-frame length of 40 with 4 sub-frames per frame is also retained from the coder structures discussed in the previous chapters. The architecture of the mixed MSE/BSD time domain CELP coder is shown in Figure 5.6.

Since the BSD depends on the initial calculation of a DFT spectra, it is necessary to avoid spectral splatter by windowing the time samples. Further, the window must be greater than a single sub-frame in length so as to fully represent pitch dynamics in the Bark spectrum. So as to cater for the majority of pitch periods, (ranging from 16 to 147 samples in 8kHz sampled speech) the window length is set to 120 samples and positioned so as to contain 60 samples from the history of the input vector and 60 new samples from the current and future sub-frame. This gives a true Bark spectral representation for the current sub-frame and avoids discontinuities caused by taking too short a sample record. The choice of 120 samples also allows efficient computation of the DFT, by use of a 128 point FFT, on a sequence with minimal zero padding.

For the BSD computation, two Bark spectra are required; one to represent the unweighted input speech and the second to represent the candidate vector generated from the codebook search. The construction of these two vectors in the time domain is now considered.

5.2.1 Construction of input speech vector, x(n)

The input speech vector x(n) is, simply, constructed from the unweighted input speech. Note that throughout the preparation of vectors for BSD calculation the LPC filters are unweighted; the perceptual weighting operation, required by the standard MSE search procedures, is replaced by the perceptually meaningful BSD computation.

The unweighted input speech vector, $x(n)$, is windowed by a Hamming window, centred on the beginning of the current sub-frame. The windowing operation is described by:

$$x(n) = s_i(M - 60 + n)w(n) \qquad for\ n = 0,1,2,...119 \quad .....(5.11)$$

where $x(n)$ is the prepared vector

$w(n)$ is a 120 point Hamming window

$s_i(n)$ is the input speech and M is the first sample of the current sub-frame

5.2.2 Construction of candidate synthesized vector y(n).

The calculation of $y(n)$ is more complex than that of $x(n)$ due to the obvious lack of 'future' synthesized speech samples. Further, the essential retention of the adaptive codebook means that its contribution to the synthesized speech must be considered in the fixed codebook search. The first section of the vector is unaffected by either of these constraints, and is constructed from the previous 60 synthesized samples $s_r(n)$ These are positioned in $y(n)$ such that:

$$y(n) = s_r(M - 60 + n)w(n) \qquad for\ n = 0,1,2,...59 \quad .....(5.12)$$

It is also useful to precalculate a second vector, $s'_i(n)$, to be used for the MSE gain calculation in the fixed codebook search. This 'pseudo' vector is

calculated by subtracting the zero vector synthesis filter response and the adaptive codebook contribution from the input speech. Thus:

$$s_i'(n) = s_i(M+n) - z(n) - \chi_a a^{60}(n) \qquad \text{.....(5.13)}$$

where χ_a and $a^{60}(n)$ are the adaptive codebook gain, and an extended 60 sample adaptive codebook contribution, respectively. The extended adaptive codebook contribution is produced by adding further samples from the codebook history to the chosen 40 point vector. In equation (5.13), *z(n)* represents the inverse LPC filter's response to a zero vector.

The BSD search is now performed with the search loop containing the minimum of calculations. First, the synthesis filter response to each codebook vector $c_q^{60}(n)$is computed:

$$s_p(n) = c_q^{60}(n) + \sum_{k=1}^{10} a(k) s_p(n-k) \qquad \text{.....(5.14)}$$

$$for\ n = 0,1,....59$$

where *a(k)* are the computed LPC coefficients for the current frame. Note that the synthesis filter has unweighted coefficients, as before, and that the code vector, although 40 samples long, is zero padded to facilitate the synthesis filter 'ringing'. These extra samples are effectively a representation of the next sub-frame, assuming its samples to be identically zero. This is clearly an approximation but, since future samples are never available it is a reasonable compromise. In practice, the technique was found to give good results.

The results of equations (5.13) and (5.14) can now be used to calculate the fixed codebook gain, both for possible transmission and correct construction of *y(n)*.

The gain χ_f^q is calculated using the familiar MSE calculation discussed in chapter 3:

$$\chi_f^q = \frac{\sum_{n=0}^{39} s_i'(n)s_p(n)}{\sum_{n=0}^{39} s_p(n)s_p(n)} \qquad \text{.....(5.15)}$$

This calculation is performed over 40 samples since the code will only represent 40 samples in the synthesised speech. The extra 20 samples are only used for the final construction of $y(n)$ which is performed as:

$$y(n+60) = \left(\chi_f^q s_p(n) + z(n) + \chi_a a^{60}(n)\right) w(n+60) \qquad \text{.....(5.16)}$$
$$for\ n = 0,1,.....59$$

The last half of the 120 sample Hamming window $w(n)$ is also applied to complete the windowing operation. Thus equations (5.12) and (5.16) fully describe the prepared vector $y(n)$, *n=0,1,...119.*

Both $x(n)$ and $y(n)$ are then passed to the BSD calculation described previously. The BSD uses 128 point FFTs to compute the magnitude spectra of the two vectors $x(n)$ and $y(n)$ and the resulting Bark spectra are then compared according to equation (5.10) to produce a measure of distortion between the two input vectors. The search is continued for all q (i.e. a codebook length of 1024 vectors) and the vector $c_q^{60}(n)$ minimising the BSD computation of (5.10) is chosen to represent the current sub-frame. The CELP process then continues for the following subframes as described in section (3.4.3).

Although the process described achieves the integration of the BSD into a CELP architecture, it is clearly non-ideal with the necessity of transforming vectors to the DFT domain for each codebook vector. The

following section considers the introduction of the fixed codebook BSD search into a Frequency Domain CELP architecture.

5.3 Incorporation of the BSD into Frequency Domain CELP

Since the BSD requires calculation of the DFT of each candidate vector, the use of a frequency domain search for the adaptive codebook remains impractical for a real-time speech coder. The adaptive codebook search is ,thus, again retained as a time domain operation.

The major problem associated with incorporation of the BSD is the production of the windowed synthesised speech vector, $y(n)$ (and its DFT $\mathbf{Y}(k)$), consisting of samples from the previous, present and future synthesised sub-frames. In the time domain this combination of vectors is performed by addition and concatenation, but in the DFT domain concatenation of equivalent time domain vectors is more complex. There are, however, computational advantages of a wholly frequency domain solution, namely that the DFT count is substantially reduced by holding the fixed codebook in the DFT domain.

The chosen solution to the concatenation problem exploits the zero padding of vectors which is already performed in the frequency domain coder (see section 4.3.2). For the BSD search, each DFT is set to have a length equivalent to the window length, which for convenience is set to 160 samples (this maintains compatibility with the initial frequency domain coder by simply doubling the transform length). Each 40 sample time domain sequence is, thus, zero padded to 160 samples.

The zero padding operation is further exploited to position the vectors for concatenation. The 'previous' vector samples nominally occupy the first 80 samples of the time domain window and these are transformed after the standard zero extension operation. The present sub-frame samples are,

however, the result of a convolution operation and will normally, also occupy the first 80 samples of the equivalent time domain vector. A characteristic of the zero padding operation is, thus, used to time shift the convolution result to the second 80 samples. Conveniently, this can be performed by negating the odd numbered coefficients of the 160 point transform. The two positioned vectors can then be concatenated by addition in the DFT domain.

The full incorporation of the BSD into the frequency domain coder is now considered.

5.3.1 Codebook search preparation and computation of X(k)

The preparation for the codebook search consists of the calculation of the 160 point transforms of the input speech vector, *x(n),* and the unweighted LPC synthesis filter's truncated impulse response, *h(n)*. The vector *x(n)* is similar to that defined in equation (5.11), and *h(n)* is zero padded from 40 to 160 samples. Hence the two series and their DFTs are:

$$\begin{aligned} x(n) &= s_i(M-80+n)w_{160}(n) \quad for\ n = 20,21,.....139 \\ &= 0 \qquad\qquad\qquad\qquad for\ n < 20,\ n > 139 \\ \mathbf{X}(k) &= \mathbf{DFT}[x(n)] \\ \mathbf{H}(k) &= \mathbf{DFT}[h(n)] \qquad\qquad for\ n,k = 0,1,2,...159 \end{aligned} \qquad .....(5.17)$$

In this case $w_{160}(n)$ describes a 160 sample Hamming window.

5.3.2 Computation of Y(k)

As for the time domain BSD coder, it is useful to produce a pseudo vector, $s'_i(n)$, consisting of the input speech after removal of the zero vector response and adaptive codebook contributions. This is simply the 160 point DFT of the time domain vector described by equation (5.13).

Thus:

$$
\begin{aligned}
& s_i'(n) = s_i(M+n) - z(n) - \chi_a a^{40}(n) \\
& \qquad\qquad\qquad\qquad \text{for } n = 0,1,....39 \\
& s_i'(n) = 0 \qquad\qquad\qquad \text{for } n = 40,41,...159 \\
& \mathbf{S_i'}(k) = \mathbf{DFT}[s_i'(n)] \qquad \text{for } n,k = 0,1,2,...159
\end{aligned}
\qquad \text{.....(5.18)}
$$

Since the initial samples of the time domain vector come from the previous two sub-frames, it is also possible to pre-construct part of the synthesised candidate vector. Further, the second half of this vector will consist partly of the zero filter response and the adaptive codebook contribution. Thus, part of $\mathbf{Y}(k)$, which we will define as $\mathbf{Y}'(k)$can be defined as:

$$
\begin{aligned}
& y'(n) = s_r(M-80+n) && \text{for } n = 20,21,...79 \\
& y'(n) = z(n-80) + \chi_a a^{60}(n-80) && \text{for } n = 80,81,82,...139 \\
& y'(n) = 0 && \text{for } n < 20, n > 139 \\
& \mathbf{Y}'(k) = \mathbf{DFT}[y'(n)] && \text{for } n,k = 0,1,2,...159
\end{aligned}
\qquad (5.19)
$$

The codebook search procedure then proceeds for each DFT fixed codebook entry $\mathbf{C_q}(k)$ (A standard codebook of 1024 vectors was used). Throughout the search procedure the DFTs are the full window length of 160 samples. Thus, each codebook DFT vector $\mathbf{C_q}(k)$ represents a time series zero extended from 40 to 160 samples. The inverse filtered codebook vector is produced by multiplication in the DFT domain such that:

$$\mathbf{F}(k) = \mathbf{C_q}(k)\mathbf{H}(k) \qquad \text{for } k = 0,1,2,...159 \qquad \text{.....(5.20)}$$

The gain term χ_f for the current code vector can then be calculated using the DFT domain MSE gain calculation described in section (4.1.4). It is

this calculation that uses the preprepared vector $\mathbf{S'_i}(k)$ prepared in equation (5.18):

$$\chi_f^{(q)} = \frac{\text{Real} \sum_{k=0}^{159} \mathbf{S}_i^*(k)\mathbf{F}(k)}{\sum_{k=0}^{159} |\mathbf{F}(k)|^2} \qquad \text{.....(5.21)}$$

The result of equation (5.20) is now phase shifted, using the technique described earlier, such that the results of the equivalent time domain convolution occupy the second half of the window. This is achieved by the minor alteration to $\mathbf{F}(k)$:

$$\begin{aligned} \mathbf{F}(k) &= -\mathbf{F}(k) && \textit{for } k \textit{ odd} \\ &\textit{unchanged} && \textit{for } k \textit{ even} \end{aligned} \qquad \text{.....(5.22)}$$

Combining the result of equations (5.21), (5.22) and the preprepared $\mathbf{Y'}(k)$ from equation (5.19), the candidate synthesised vector for the BSD computation can be computed as:

$$\mathbf{Y}(k) = \left(\chi_f^{(q)}\mathbf{F}(k) + \mathbf{Y'}(k)\right) \otimes \mathbf{W}(k)$$

$$\textit{for } k = 0,1,2,\ldots 159, \qquad \text{.....(5.23)}$$

(⊗ indicates circular convolution)

In (5.23) $\mathbf{W}(k)$ is the DFT of the 160 point Hamming window $w_{160}(n)$. The Hamming window is defined in the discrete frequency domain by just three non-zero coefficients. These are at -1,0,1 and have values of -0.23,0.54,-0.23 respectively. The circular convolution operation in equation (5.23) can thus be reduced to a weighted addition of the original and two shifted versions of the bracketed series.

Finally, the BSD measure is performed as described in section (5.1), excepting the requirement to calculate the DFTs of the input sequences.

Also, for the DFT domain search, all BSD computations take place on 160 point transforms.

5.4 Comparison of MSE and BSD searches.

Figures (5.7) and (5.8), show the values of the MSE and BSD measures for identical codebooks for searches of three sub-frames. The MSE search results are for the maximisation of the expression of equation (3.37) discussed in chapter 3. Thus, in comparing the search processes it should be noted that the MSE search seeks to maximise the error term, while the BSD search is a minimisation process. It is also clear, from the graphs, that the range of the MSE and BSD calculations are significantly different. These search comparisons show that the codes chosen using the BSD are unrelated to those chosen by an MSE search. In fact, the codes chosen by one search technique generally do not score highly using the other measure.

The search results contrast with informal listening tests that suggest that for many speakers the BSD gives preferable speech quality over the normal MSE search. Full objective measure results for the BSD CELP schemes are considered in the next section.

In complexity terms, BSD searched, time domain CELP requires some eight times as many multiplication operations as a standard MSE search. Exact complexity figures are difficult to determine due to the power law operation included in the computation of the BSD, but, it is clear that real-time implementation of BSD searched CELP would currently be impractical. Future increases in processor power should, however, make the use of such perceptual measures a practical proposition.

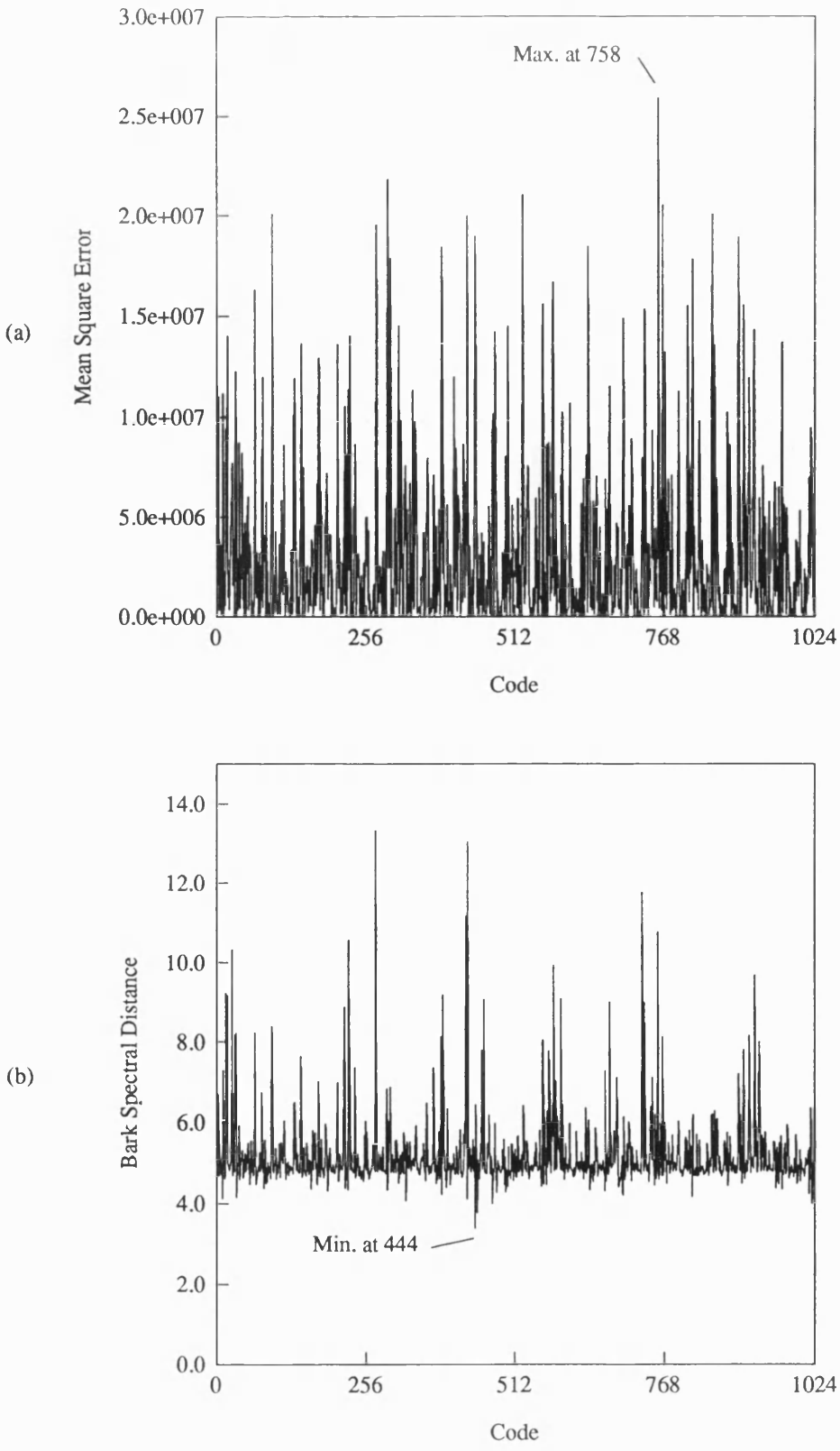


Figure 5.7: Comparison of (a) MSE and (b) BSD search results. Note that the BSD search selects the code with the lowest BSD, while the MSE search selects the code with the highest value.

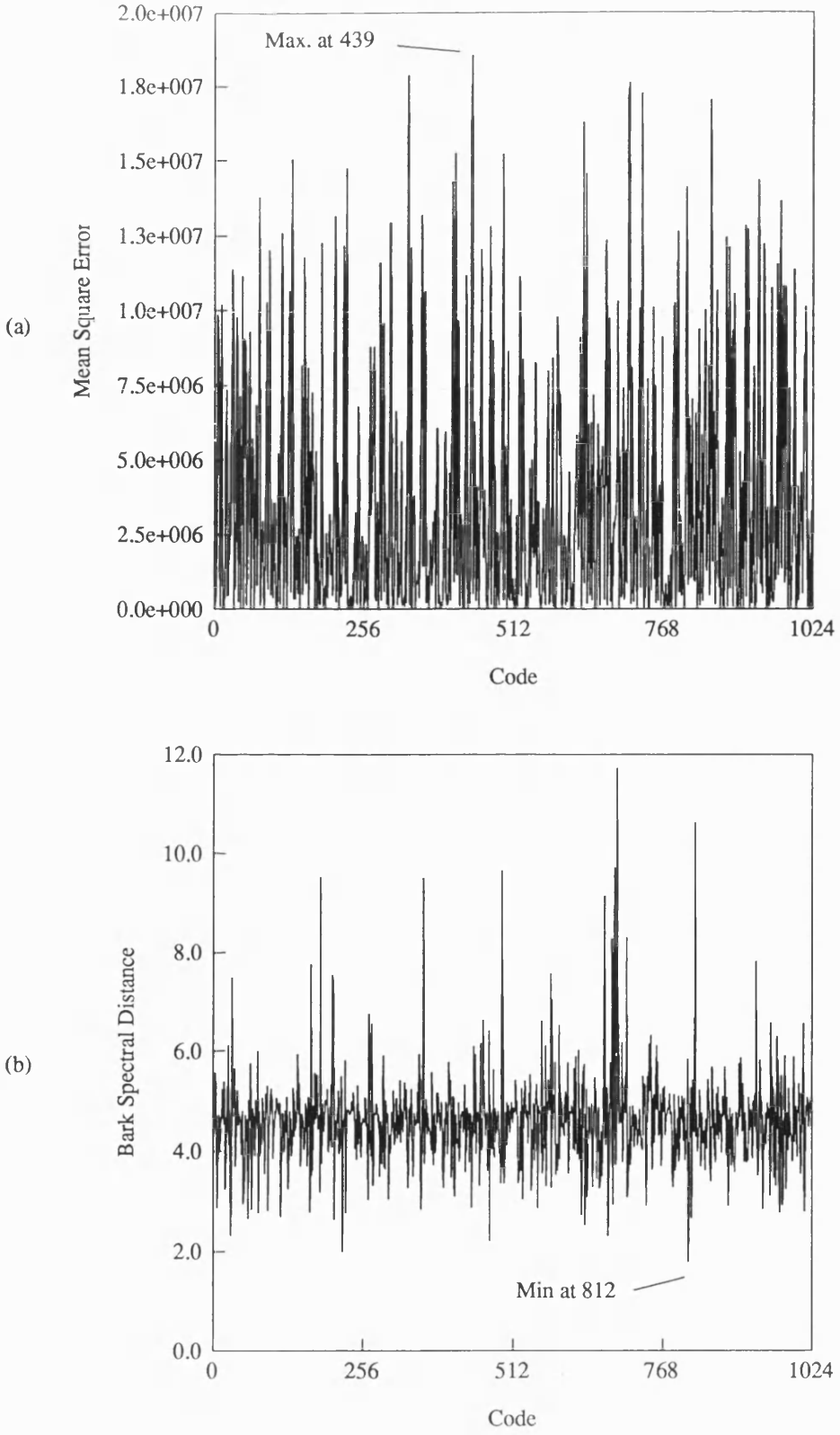


Figure 5.8: Comparison of (a) MSE and (b) BSD search results. Note that the BSD search selects the code with the lowest BSD, while the MSE search selects the code with the highest value.

5.5 Results of Time and Frequency Domain BSD CELP

Bark domain searched CELP would not be expected to produce good results for the time domain objective measures described in section (3.6.1). However, the Cepstral Distance measure compares the smoothed spectra of the input and synthesised speech and could be expected to give meaningful results for Bark domain coding.

Results for all of the objective measures are shown in the bar charts of Figure 5.9. For comparison, typical Time Domain CELP results are also shown on each chart and, unsurprisingly, the two sets of SNR results indicate that Time Domain CELP significantly outperforms the BSD CELP. Time Domain CELP minimises the time variance between the input and synthesised speech vectors while the Bark domain search matches the speech spectra. Thus, the time-domain objective measure results are, effectively, confirming the search techniques used by each coder.

The CD measure indicates that the Bark coded speech has a better spectral match than the standard Time-Domain CELP. However, since the range of CD measures is low these margins, alone, should not be considered significant.

Finally, it is noted that for all results, bar one, the Time-Domain Bark searched CELP architecture out-performs that searched in the Frequency domain. The margin is small, but possible reasons are the extended DFT and window lengths used.

Informal subjective listening tests suggest that the Bark coded speech is superior to Time-Domain CELP for most speakers. Bark coded speech tends to sound more natural and less harsh than standard CELP schemes. Comparative coded speech waveforms are shown in Figure 5.10.

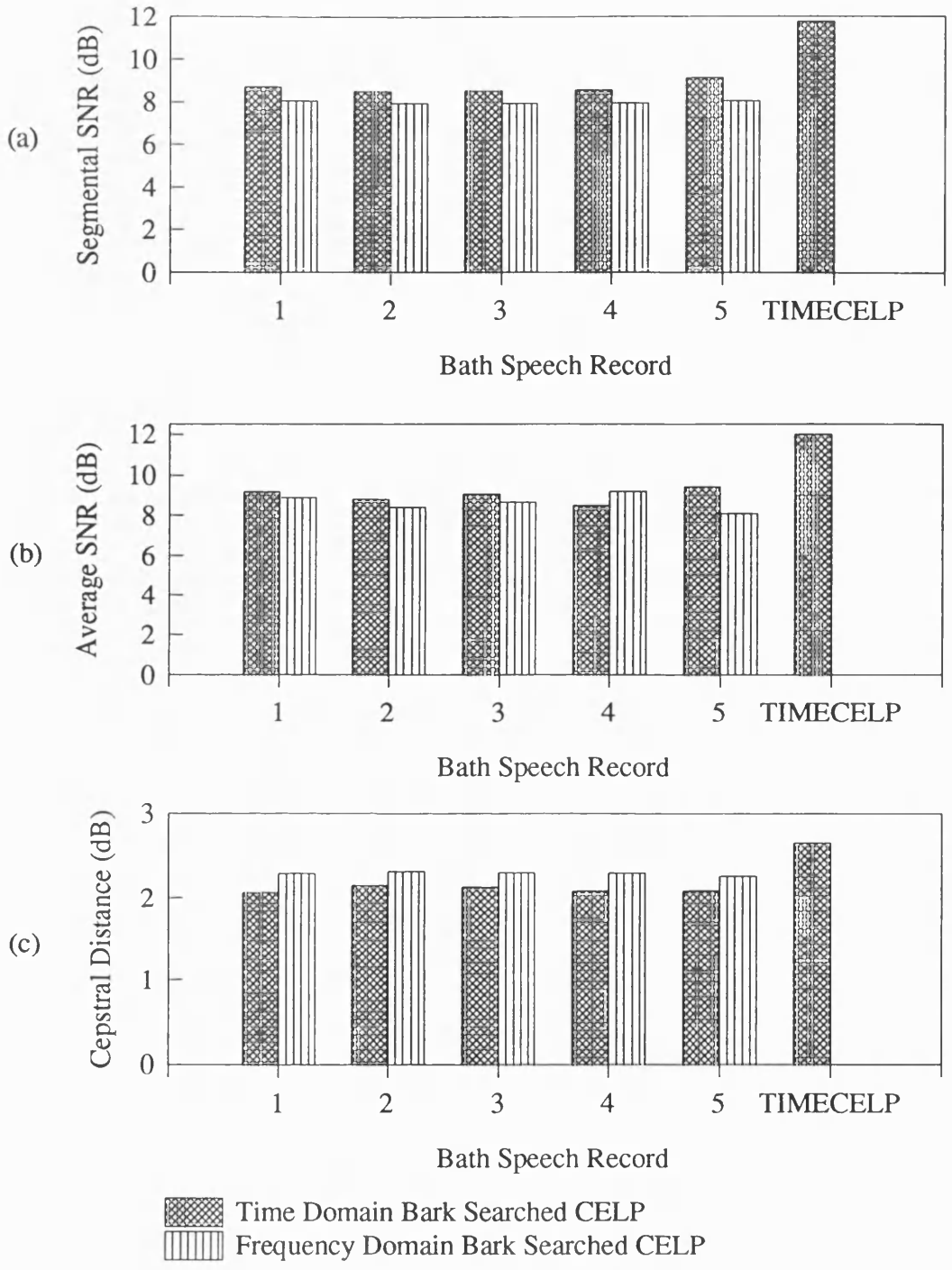


Figure 5.9: Objective Measure results for Bark domain searched CELP coding. (a) Segmental SNR (SEGSNR), (b) Average SNR (Av. SNR), and (c) Cepstral Distance (CD).

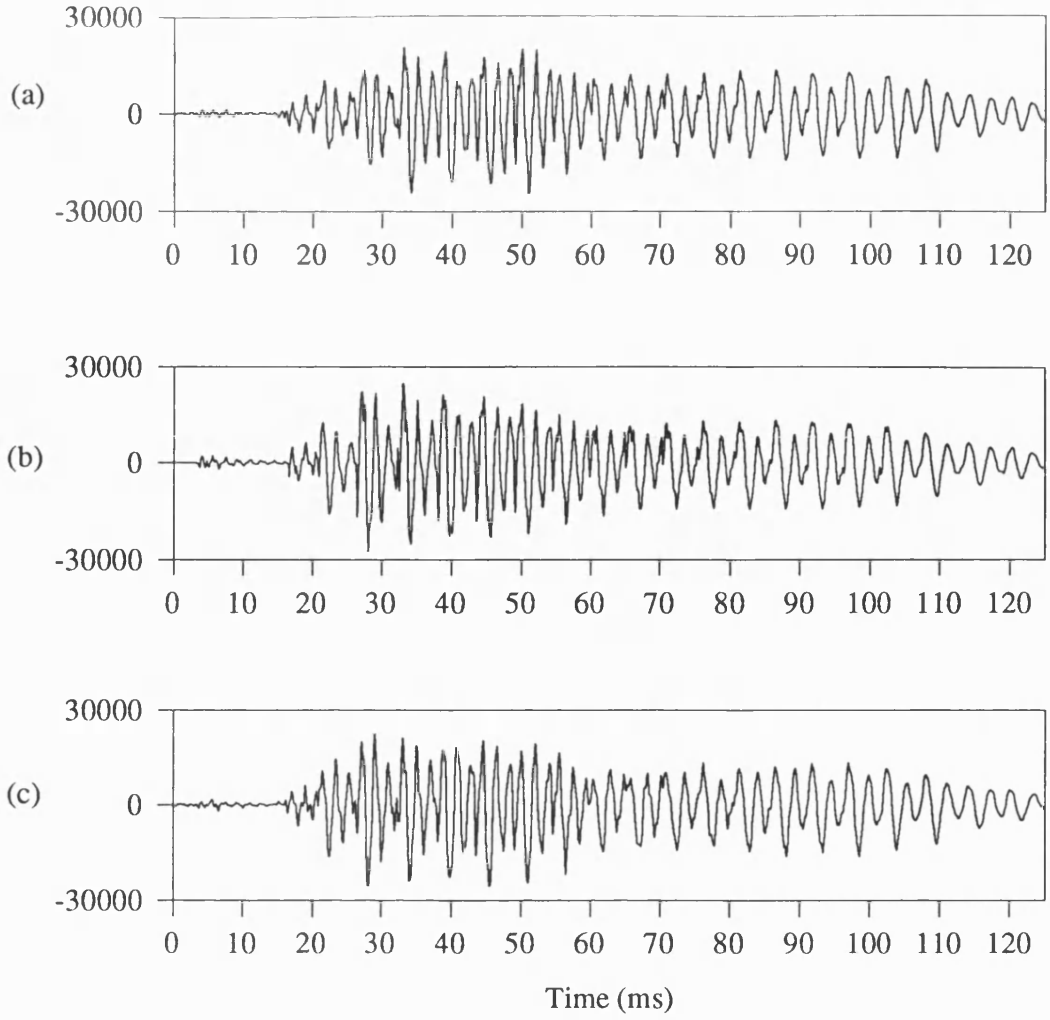


Figure 5.10: Comparison of standard MMSE CELP output (a) and that of the BSD search CELP coder (c). For reference the input speech (the word 'oak') is also shown (b).

5.6 Reduced spacing of Bark domain filters

In the previous sections, for both time and frequency domain configurations, the critical band filter functions were spaced at 1 Bark intervals. While this interval, gives adequate coverage across the speech bandwidth, the ear would actually have many more 'effective' critical band filters spaced along the basilar membrane. In the paper by Jenison et al. [2] which considers a cat ear, 128 auditory nerve fibre channels are used. In these terms, the sixteen filters cannot be a very realistic model of the perceptual behaviour of the ear. However, the calculation complexity

of the critical band filtering is high, making the use of 128 filters impractical. This is especially true if the BSD were to be used as a real-time search measure. However, in simulation it was possible to consider an increased filter density and, for simplicity, two cases were considered:

1. The number of critical band filters was doubled such that the filters are 1/2 Bark spaced across the speech bandwidth. This gives a total of 31 filters.
2. The number of critical band filters was quadrupled such that the filters are 1/4 Bark spaced. This results in a total of 61 filters covering the 4kHz bandwidth.

These changes are simply applied to the coders by altering the convolution operation of equation (5.4), such that the sampled sequences $\mathbf{D}(i)$, $\mathbf{F}(i)$ and $\mathbf{Y}(i)$ are sampled at 1/2 and 1/4 Bark intervals. In practice, this means that there is a doubling and quadrupling of the effective 'sampling rate' of each Critical Band filter function.

With the increased number of filter results, the BSD is computed over an increased number of points, such that the BSD computation, described by equation (5.10), becomes:

$$\mathbf{BSD}^{(k)} = \sum_{i=1}^{31} \left[\mathbf{L}_x^{(k)}(i) - \mathbf{L}_y^{(k)}(i) \right]^2 \qquad \text{.....(5.24)}$$

for the 1/2 Bark case and,

$$\mathbf{BSD}^{(k)} = \sum_{i=1}^{61} \left[\mathbf{L}_x^{(k)}(i) - \mathbf{L}_y^{(k)}(i) \right]^2 \qquad \text{.....(5.25)}$$

in the 1/4 Bark case. The result of these changes is an overall increase in the resolution of the BSD calculation.

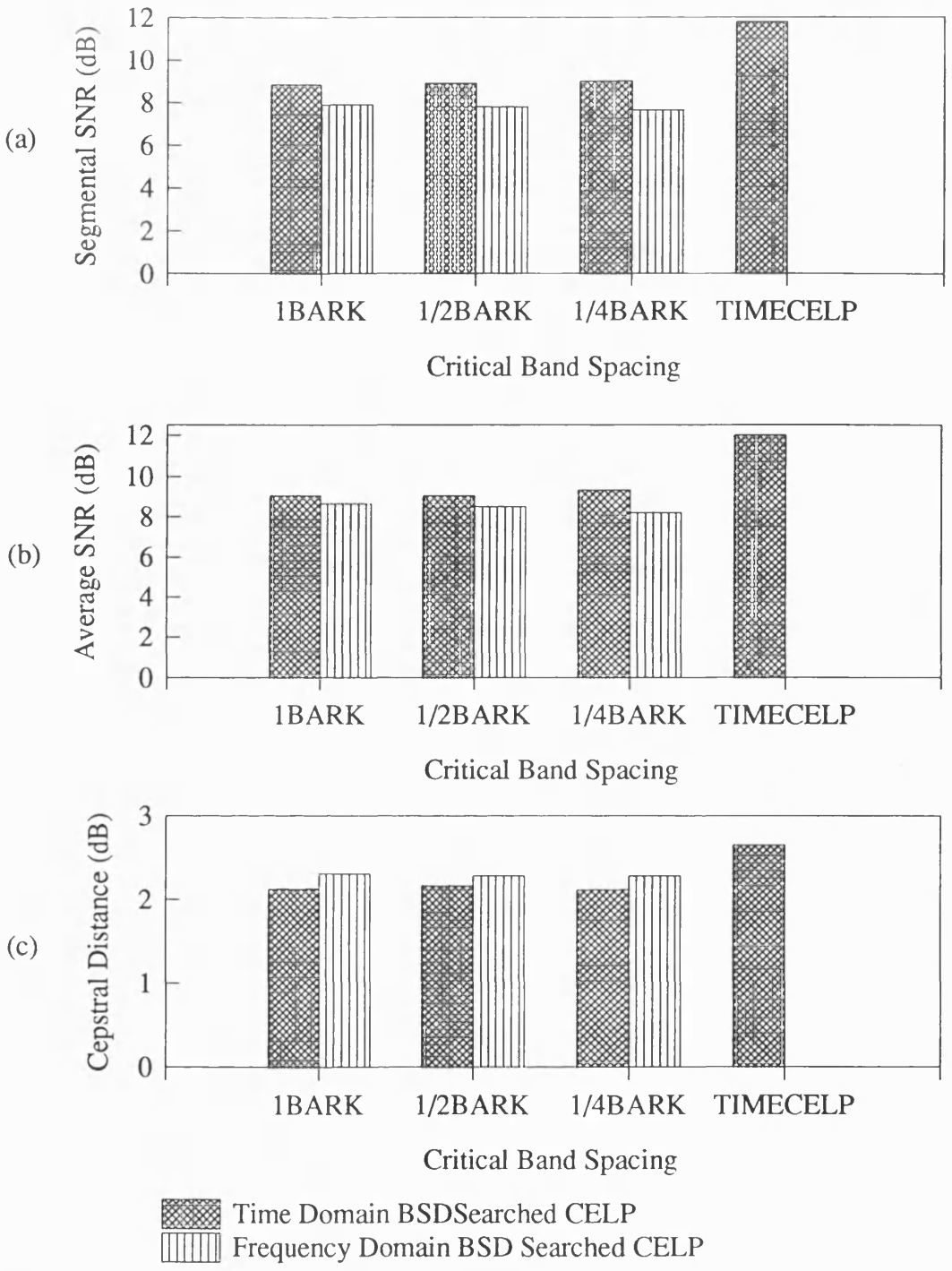


Figure 5.11: Bar charts showing the Objective performance of BSD CELP using 1, 1/2 and 1/4 Bark spaced Critical Band filters.

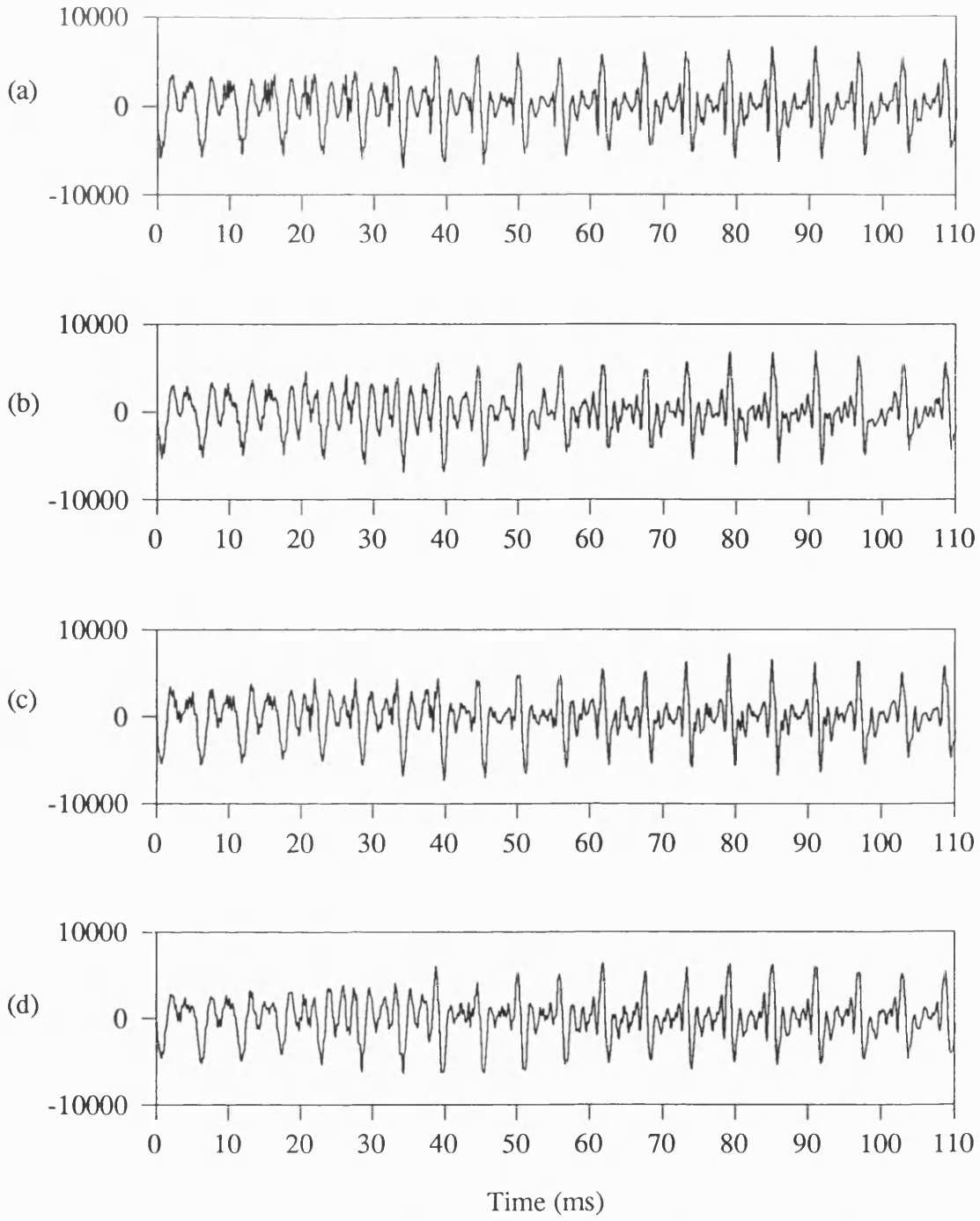


Figure 5.12: Waveforms of input speech (a) coded using a Time Domain CELP coder employing (b) a 1 Bark filter spacing BSD measure, (c) a 1/2 Bark filter spacing BSD measure and (d) a 1/4 Bark filter spacing BSD. measure.

The improved coders were used to encode Bath Speech Record 3. The time domain measures resulting are shown in the bar charts of Figure 5.11. These show that there is little objective difference between the speech coded with 1 Bark filters and that coded with the improved resolution. The CD measure, however, suggests that BSD CELP achieves an increased level of spectral matching than standard Time Domain CELP and that this improves with increased Critical Band filter density. Informal listening tests confirm this result: The increase in resolution to 1/2 Bark makes the coded speech sound more natural, with a further slight improvement on the increase to 1/4 Bark spacing. Figure 5.12 shows a comparison between speech coded with 1 Bark, 1/2 Bark and 1/4 Bark filters.

The waveforms show that the increase in resolution of the BSD improves the detail of the coded speech. This corresponds with the audible improvement.

5.7 The BSD as an Objective measure

Since the BSD is a perceptually meaningful comparison of two speech waveforms it can be used in a stand-alone mode as an objective measure of coder performance. For objective measure calculations the BSD is computed on a frame-by-frame basis and a mean taken over the speech record. This is a similar approach to that used for the other measures, described in chapter 2.

A further complication in the use of the BSD as a measure is described by Wang [4]. The BSD is inherently dependent on absolute values of the speech waveform. Thus the sensitivities of A/D converters in the original sampling processes will alter BSD results.

The BSD is therefore normalised by dividing by the average Bark energy in the signal:

$$E_{Bark} = \underset{k}{Ave} \sum_{i=1}^{N} \left[\mathbf{L}_x^{(k)}(i)\right]^2 \qquad \text{.....(5.26)}$$

Thus the normalised BSD measure becomes:

$$\mathbf{BSD_N} = \frac{\underset{k}{Ave} \sum_{i=1}^{N} \left[\mathbf{L}_x^{(k)}(i) - \mathbf{L}_y^{(k)}(i)\right]^2}{E_{Bark}} \qquad \text{.....(5.27)}$$

Wang [4] recommends the use of a segment/frame length of 80 samples, however in the following results both 160 sample and 80 length windows are used. These correspond directly with BSD computations performed in the Time and Frequency Domain BSD search coders respectively.

Since the BSD is a spectral measure, similar in nature to the CD, a test on its performance was made using the 'essential' coefficient coder results from section (4.5). These introduce an increasing degree of spectral distortion as the number of coefficients is reduced. When a high number of 'essential' coefficients are retained the coder produces exceptionally high speech quality. Results for the BSD on these coder records are shown in Figure 5.13 and are directly comparable with those for the other objective measures shown in Figures 5.9 and 5.11.

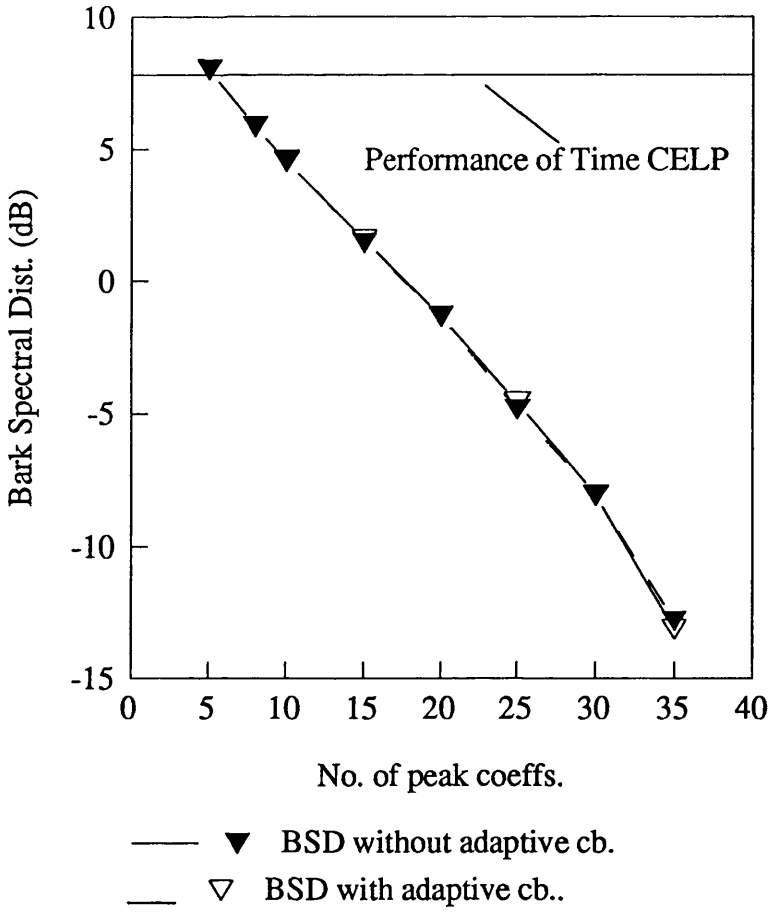


Figure 5.13: Results for the BSD (window length 80) measure on speech records generated by the 'essential' coefficient coder described in section (4.5).

These curves confirm the previous results that just 5 coefficients need be used to better the performance of Time Domain CELP in the 'essential' coefficient coder architecture.

A further set of BSD measure results were computed for the various Time Domain BSD search records. Results for these and comparative overlapped codebook Frequency domain CELP architectures (see section (4.4)) are shown in the bar chart of Figure 5.14.

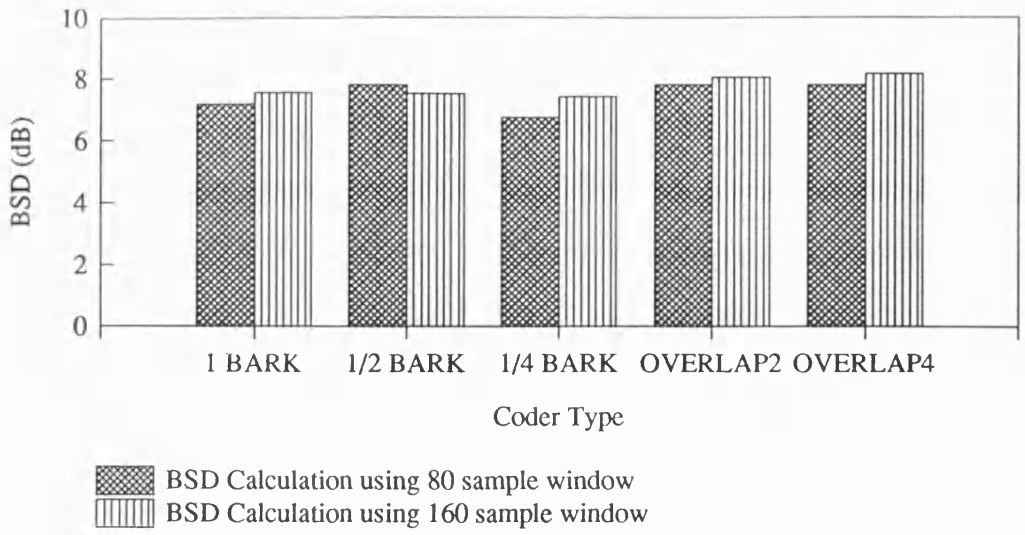


Figure 5.14: Results of the BSD objective measure for BSD searched CELP using 1, 1/2, and 1/4 Bark filter spacings. For comparison Overlapped Frequency domain CELP results with overlaps of 2 and 4 are also shown.

From the bar chart it can be seen that the length 80 and length 160 windows give similar results and the Bark domain search is seen to outperform the standard MSE Frequency Domain CELP coders. It can also be seen that the 1/2 and 1/4 Bark searches show progressively reducing BSD measures. This contrasts with the SEGSNR and AV.SNR objective measures that suggest a deteriorating performance with increased Bark resolution, but confirms the results of the CD, discussed previously.

5.8 Conclusions

Standard CELP searches include primitive perceptual effects by the use of a weighting filter; this chapter has considered a CELP search including a more complex and realistic perceptual measure. The BSD was incorporated into both Time Domain and DFT Domain searched CELP architectures. The latter offer a more efficient computation by avoiding the necessity for DFT transformations of each convolved code vector.

While SEGSNR and AV.SNR objective measures were shown to produce invalid results for BSD CELP coded speech, the Cepstral Distance measure indicates that BSD searched CELP produces speech of a higher perceptual quality than MSE searched CELP. These results were confirmed in informal listening tests, which suggest that BSD coded speech sounds less harsh and more mellow than that produced by standard MSE CELP schemes.

Further improvements to speech quality were produced by increasing the number of Critical Band filters used in the BSD computation. The use of more filters, covering the same speech bandwidth, takes the BSD model closer to the 'ideal' perceptual model of the auditory system and speech coded with the increased resolution measure was found to have an improved degree of naturalness.

The use of the BSD as a perceptually meaningful, objective measure was also considered. By use of previously coded speech it was shown that the BSD measure performs well for spectrally distorted speech. The BSD was also used to confirm the results of BSD CELP coded speech. In general, the BSD appears to track human perception more reliably than previous objective measures.

In summary, the BSD is a perceptually meaningful, speech measure which can be used to improve the perceived quality of CELP speech coders. This extends the principles of perceptual weighting included in the standard CELP coder. At the expense of further increases in computational complexity, the BSD can be improved by increasing Critical Band filter density. This makes the BSD a closer approximation to the human auditory processes and produces further improvements in the perceived quality of the coded speech.

5.9 References

[1] B. C. J. Moore, "Introduction to the Psychology of Hearing," *The Macmillan Press Ltd.*, 1977.

[2] R. L. Jenison, S Greenberg, K. R. Kluender and W. S. Rhode, "A composite model of the auditory periphery for the processing of speech based on the filter response functions of single auditory-nerve fibers," *J. Acoust. Soc. Am.*, Vol. 90, No. 2, pp. 773-786, Aug. 1991.

[3] S. Wang, A. Sekey and A. Gersho, "Auditory Distortion Measure for Speech Coding," *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Proc.*, pp. 493-496, May 1991.

[4] S. Wang, A. Sekey and A. Gersho, "An Objective Measure for Predicting Subjective Quality of Speech Coders," *IEEE J. on Sel. Areas in Comms.*, Vol. 10, No. 5, pp. 819-829, June 1992.

[5] S. Quackenbush, T. P. Barnwell and M. Clements, "Objective Measures of Speech Quality," *Prentice-Hall*, 1988.

[6] E. Zwicker and E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency," *J. Acoust. Soc. Am.*, Vol. 68, No. 5, pp. 1523-1525, Nov. 1980.

[7] E. Zwicker, "Subdivision of the Audible Frequency Range into Critical Bands (Frequenzgruppen)," *J. Acoust. Soc. Am.*, Vol. 33, No. 2, Feb. 1961.

[8] A. Sekey and B. A. Hanson, "Improved 1-Bark bandwidth auditory filter," *J. Acoust. Soc. Am.*, Vol. 75, No. 6, pp. 1902-1904, June 1984.

[9] R. A. W. Bladon and B. Lindblom, "Modeling the judgment of vowel quality differences," *J. Acoust. Soc. Am.*, Vol. 69, No. 5, pp. 1414-1422, May 1981.

[10] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Am.*, Vol. 87, No. 4, pp. 1738-1752, April 1990.

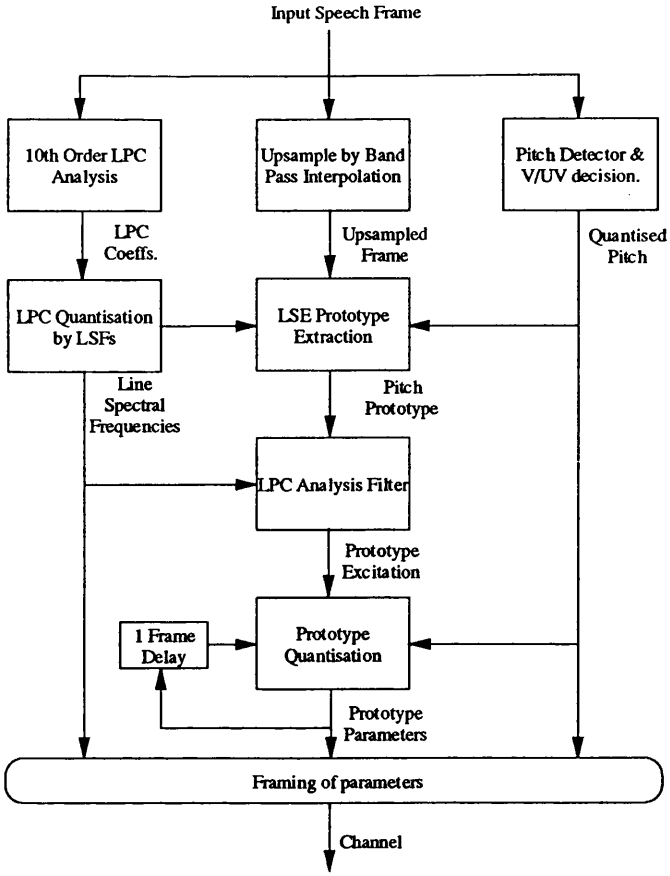
[11] D. W. Robinson and R. S. Dadson, "A re-determination of the equal-loudness relations for pure tones," *British Journal of Applied Physics*, Vol. 7, pp. 166-181, May 1956.

Chapter 6: Prototype Waveform Coding

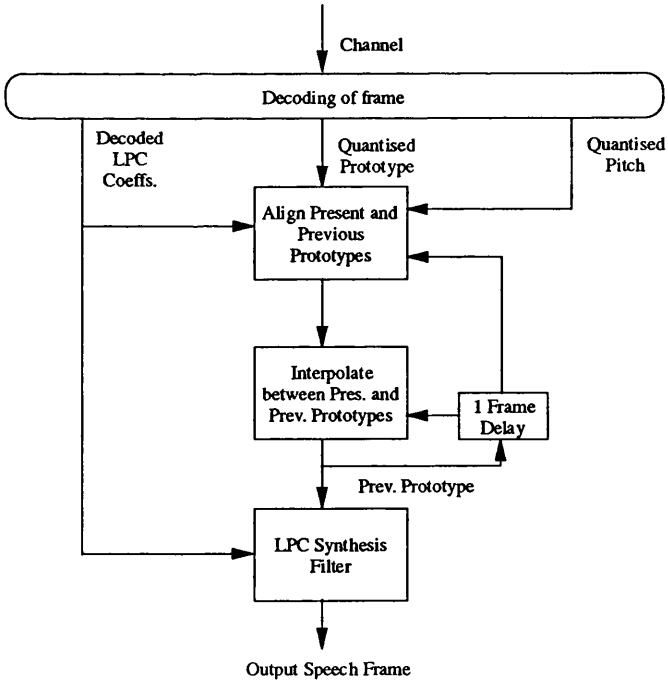
The speech coders discussed in the previous chapters of this thesis have not made a distinction between voiced and unvoiced speech frames. The coding of voiced speech requires a periodic excitation source (provided in CELP by the pitch predictor) which remained when unvoiced sections of the speech were coded. In sub 3.4 kbit/s coding, where transmission bits are at a premium, it is desirable to code the pitch information only during voiced frames. A more appropriate excitation source, than a gaussian codebook, would also be desirable for these frames. This chapter addresses these problems by using a new prototype waveform architecture for voiced frames, while retaining a CELP algorithm for unvoiced frames.

For voiced frames a single 'residual prototype' is selected to represent a voiced section of 25ms. The prototype is a small sample segment which is repeated to form the excitation for the whole speech frame. Prototypes are interpolated across the frame to provide a smooth amplitude and harmonic behaviour. Two coding schemes for the prototypes are discussed; a pitch harmonic / sub-band scheme operating in the DFT domain, and a codebook based time domain technique. Unvoiced frames are coded using a standard CELP architecture excluding the Adaptive Codebook search. The overall bit rate using either of the voiced frame coding algorithms is shown to be sub 3.2kbit/s for good communications quality speech.

Figure 6.1 shows the architecture of the voiced frame coder, and the elements of this coder are now discussed. Following the discussion of the prototype coder, the combination of this technique with a CELP coder, to form a speech coder operating at sub 3.2kbit/s, is considered.



(a) Prototype Waveform Transmitter Structure.



(b) Prototype Waveform Receiver Structure

Figure 6.1: The Prototype waveform encoding technique for voiced speech frames.

6.1 Pitch Determination.

The derivation of pitch synchronous residual prototypes from the input speech requires a reliable method of pitch determination. While the LTP techniques, considered previously, have been suitable for operation in a CELP type architecture, the technique used here operates open-loop on the input speech.

Pitch determination was an important part of the early speech vocoders and the technique described here is a progression of the technique developed by Rabiner et. al. [1][2] in hardware. The algorithm can be summarised as :

1. Filter the input speech signal using a 65 tap lowpass FIR filter with a 900Hz 3dB cut-off frequency.
2. A 300 (37.5ms) sample section centred on the current 200 sample (25ms) frame is selected. Note this results in the pitch processing frames overlapping by 50 samples (8ms)
3. The maximum amplitude encountered at both of the 100 sample end segments is calculated and a clipping level of 80% of the minimum of these two values is set.
4. Using the clipping level the section of speech is centre clipped
5. The autocorrelation function of this centre-clipped signal is calculated for the range of expected pitch values (16 - 147) and the point of the maximum autocorrelation is considered to indicate the pitch value for the segment.
6. The maximum autocorrelation is the normalised to the zeroth autocorrelation value to give a voiced unvoiced decision. If the normalised value exceeds 0.28 the frame is declared voiced.

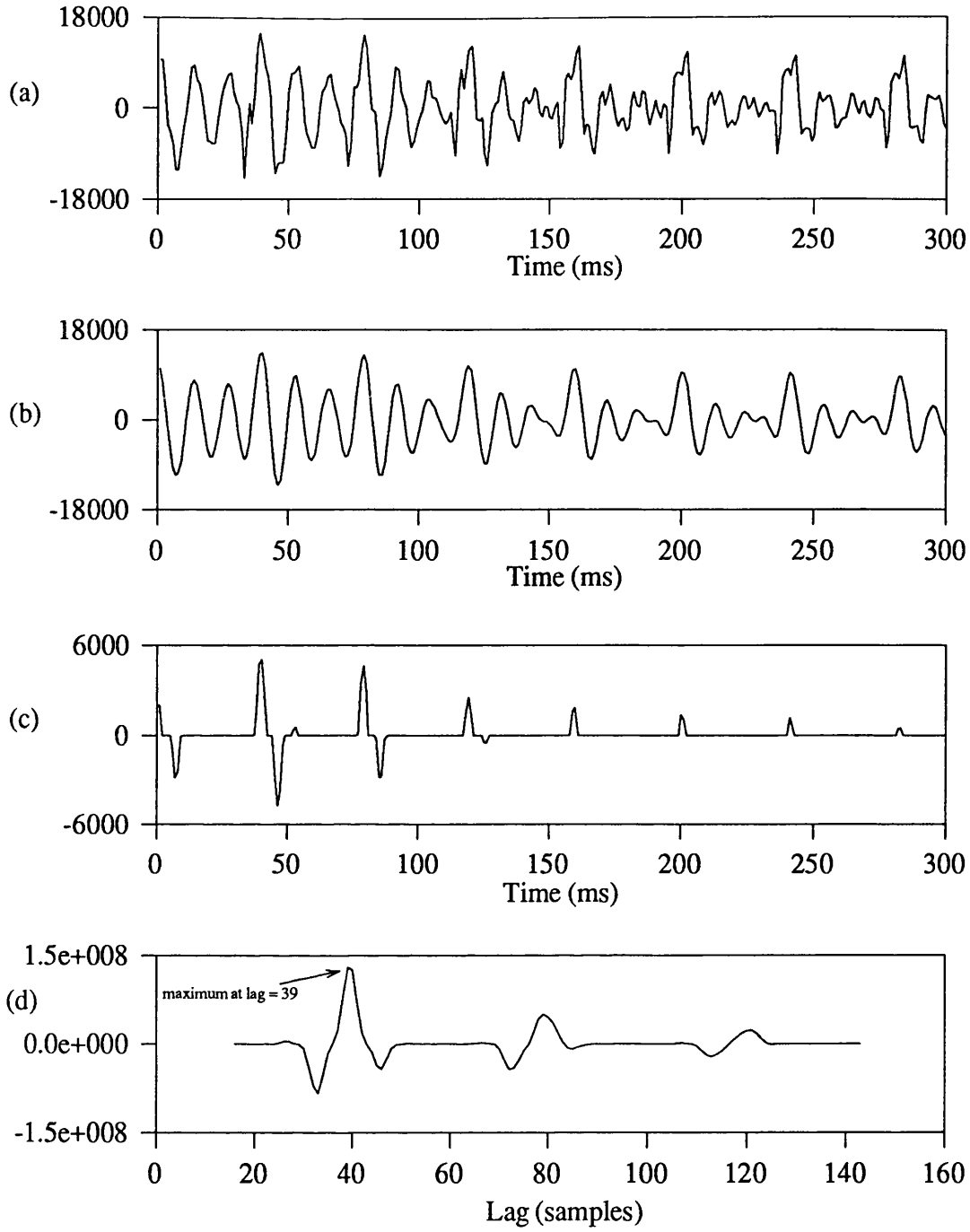


Figure 6.2: The operation of the pitch determination algorithm on a male voiced speech segment. The input speech (a) is low pass filtered (b) and then centre clipped (c). The lag corresponding to the maximum autocorrelation of (c) is then declared the pitch of the speech segment (d).

Figure 6.2 shows the basic stages of the algorithm as performed on a section of male voiced speech, and while the details of this algorithm are discussed fully in [1][2], a number of points merit clarification.

The centre clipping operation (Fig 6.2 (b)), and the choice of a clipping level, are performed such that the pitch decision is not distorted by transitional events. These events could be extraneous peaks in the autocorrelation calculations and fast amplitude changes in the input speech. The distortion is prevented by taking the minimum of the two end segment maxima and clipping to the 80% level. This level was chosen after experimentation by the author.

The voiced/unvoiced threshold (0.28) is based on that of Rabiner [1] but reduced slightly to weight the decision in favour of voiced frames. This bias was considered preferable, since a voiced frame coded using a gaussian excitation model is likely to produce more unpleasant auditory distortion than an unvoiced frame with added periodicity. The use of the voiced/unvoiced decision will be considered further in section 6.7.

6.2 Prototype Extraction.

The prototype extraction technique can be divided into three distinct processes:

- Interpolate the input speech frame.
- Extract a prototype from the input speech.
- Calculate the 'residual prototype' by LPC analysis.

The technique is further described in the block diagram of Figure 6.3 and each process is now detailed.

6.2.1 Interpolation

Extraction of a prototype from a given voiced frame is performed using a Least Squares Error calculation between a concatenated repeated prototype and the input speech frame. The calculation is performed for all possible prototypes within an interpolated speech frame and, for this

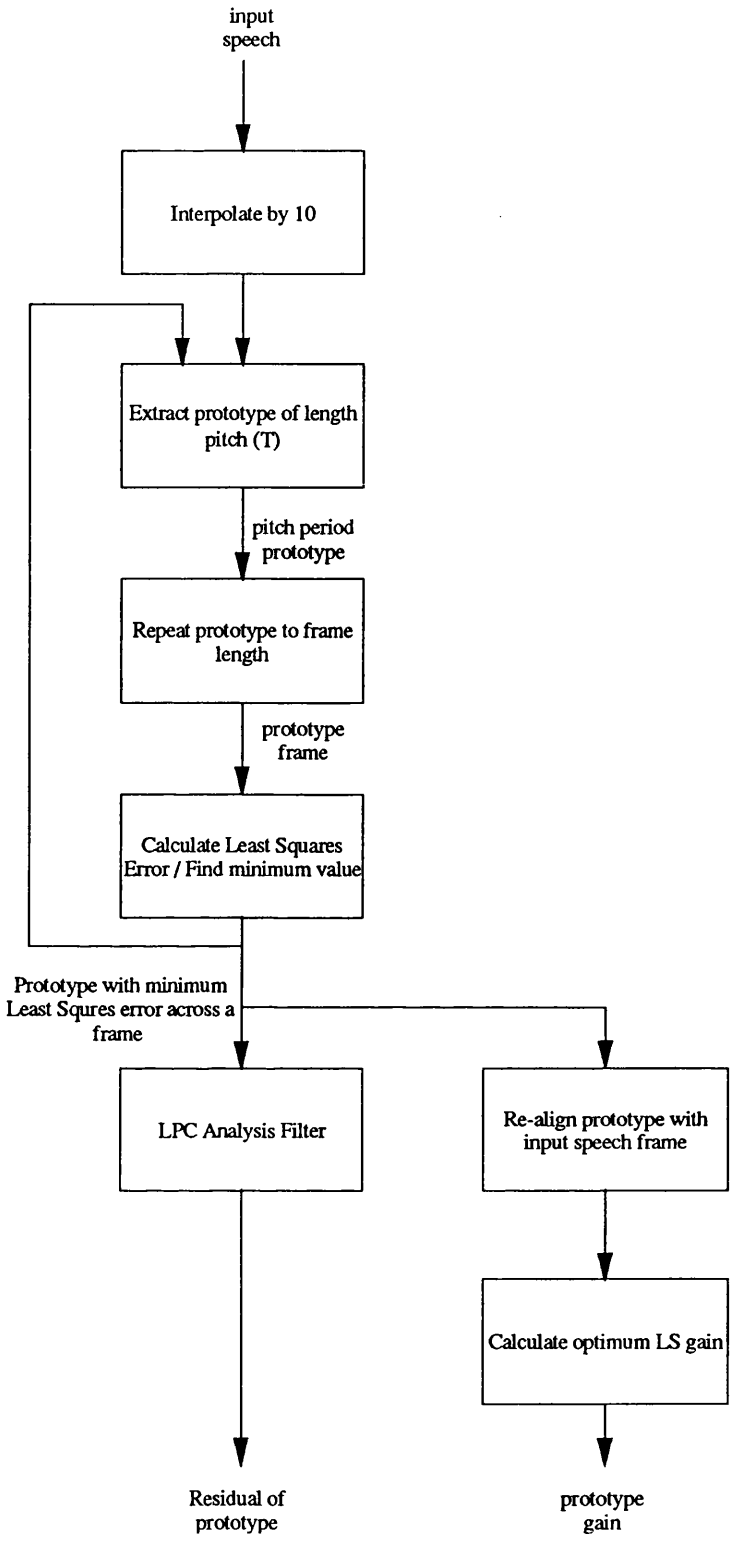


Figure 6.3: Block Diagram describing the selection of a prototype from the input speech frame by Least Square Error calculation.

purpose, the input speech is interpolated by a factor of 10 giving an effective sampling rate of 80kHz.

The interpolation process is performed by up sampling the input speech by a factor of 10 (inserting nine zero samples between each 8kHz sample) and then bandpass filtering through an interpolation filter [3]. The ideal characteristic of this filter is described by :-

$$h(k) = \frac{\sin(\pi k / L)}{\pi k / L}, \quad k = 0, \pm 1, \pm 2, ... \qquad .....(6.1)$$

For a practical implementation of such a filter the ideal filter characteristic must be windowed and a suitable window for speech is a Hamming window [3]. For practicality it was necessary to limit the number of filter coefficients to 641 resulting in the impulse and frequency responses of Figure 6.4.

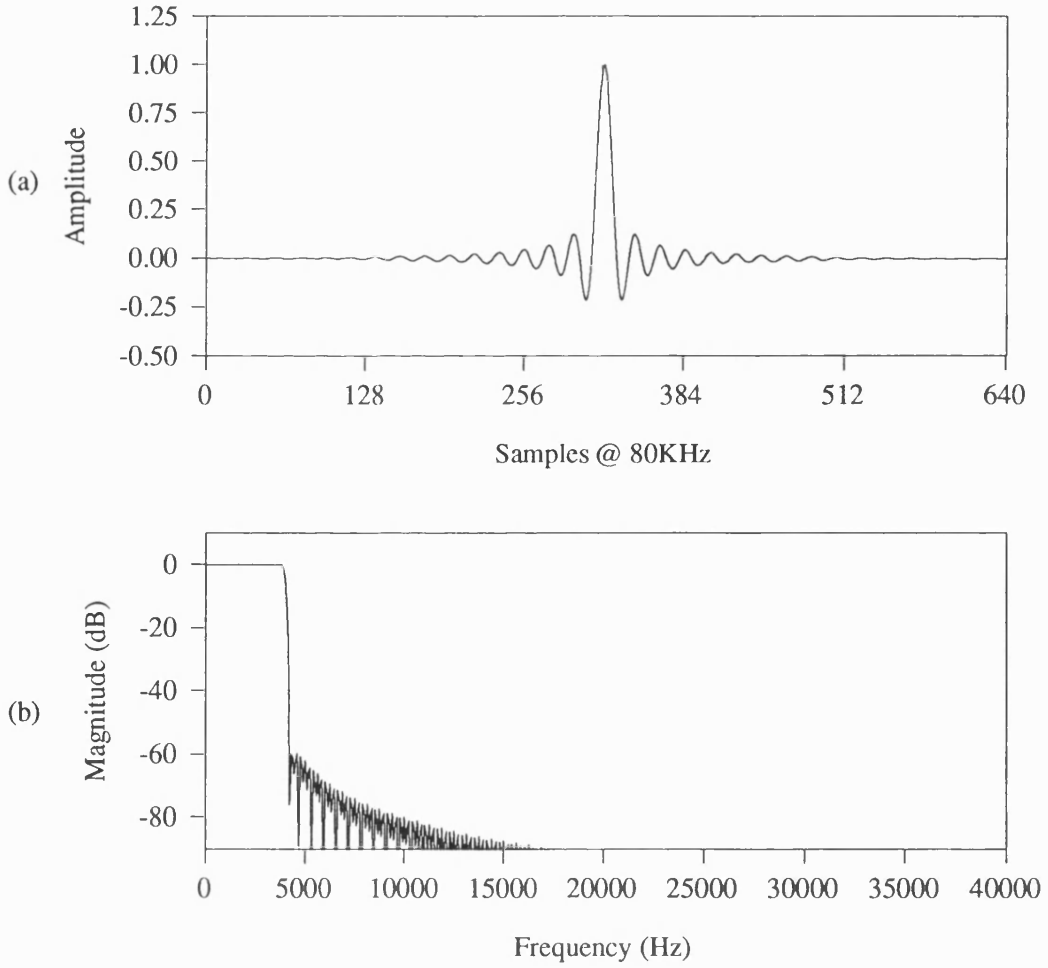


Figure 6.4: (a) Impulse response of 641 coefficient interpolating filter and (b) its spectral shape.

Due to the nature of the initial up sampling the interpolating filter lends itself to a polyphase implementation whereby only one tenth of the filter tap multipliers need be considered for a given filter output sample. This structure can be efficiently implemented [3].

One difficulty with the interpolation process is that there is an inherent group delay of 32 samples (at 8kHz). Without compensation this would result in prototypes being derived from five sixths of the current frame and one sixth of the previous frame. This was considered undesirable and the interpolation filter input is thus taken 32 samples ahead of the start of the current frame. This results in a full, interpolated version of the current frame being available for prototype extraction.

6.2.2 Prototype Derivation

Prototypes are derived from the interpolated frame by extracting τ samples from a point *start* in the interpolated frame. The prototype is sampled at the 8kHz rate such that the prototypes *p(n)* are defined as:

$$p(n) = s_i(start + n*10) \quad n = 1,.....,\tau \qquad .....(6.2)$$

This 'pitch period prototype' is then repeated to a frame length L_f (in this case 200 samples) to produce an 'extended prototype frame':

$$p_f(n) = p\left(\left(\frac{start}{10} + n\right) \bmod \tau\right) \quad n = 1,.....,L_f \qquad .....(6.3)$$

The prototypes within the prototype frame are arranged to be synchronous with the base prototype in terms of its frame position. The process is described graphically in Figure 6.5 The mean square error E_p between the input speech frame s and the extended prototype frame p_f is then calculated:

$$E_p = \sum_{n=1}^{L_f} \left(p_f(n) - s(n)\right)^2 \qquad .....(6.4)$$

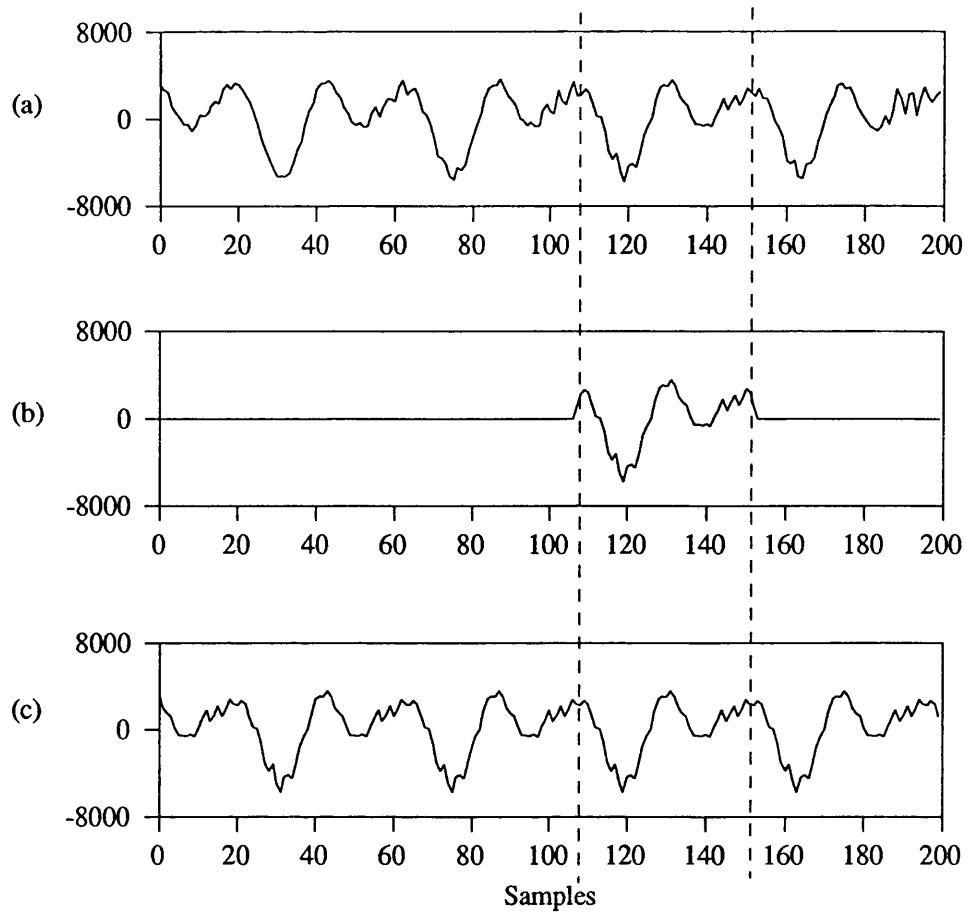


Figure 6.5: The extraction of a prototype (b) from an input voiced speech frame (a) and the synchronous repetition of the prototype to form an extended prototype frame (c). The mean square error calculation is then performed between the input speech frame and the derived prototype frame.

This calculation is repeated for all possible prototype starting points (between 0 and $L_f - \tau$) and the prototype minimising the value of E_p is chosen as the prototype to represent the current voiced frame. A gain adjustment was also included in the prototype extraction process since in certain frames it was found that a low amplitude prototype was sometimes chosen. This is partly due to the frames not being synchronous with the transitions of pitch prototypes in the input speech i.e. a frame may contain more than one possible prototype. While the selected prototype, when repeated, produced an acceptable representation of the input speech the amplitude required adjustment. A gain term was, thus,

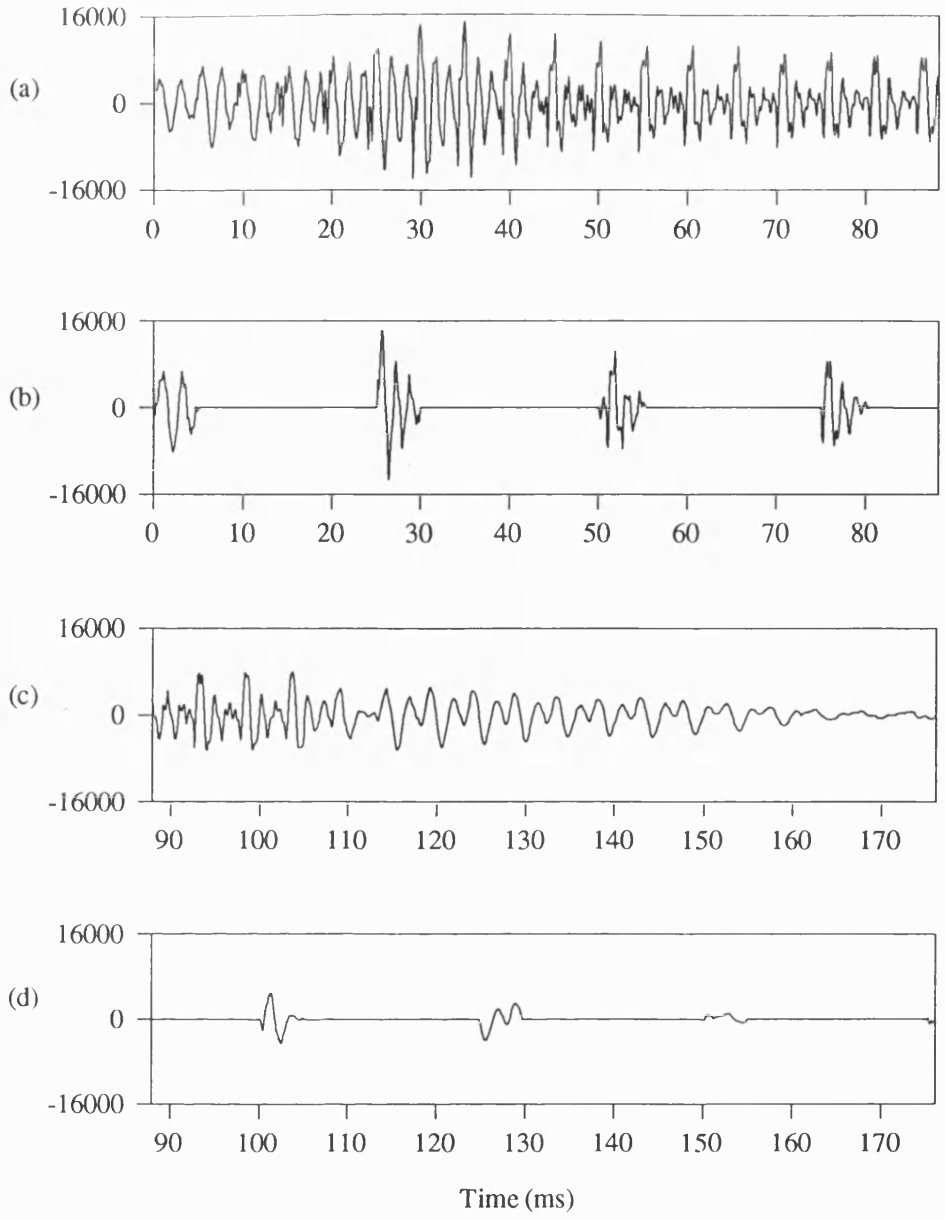


Figure 6.6: Prototypes extracted (b)(d) from successive frames of voiced speech from a female speaker(a)(c).

introduced. G_p is calculated using the standard least squares gain optimisation over the central 100 sample section of the input speech frame:

$$G_p = \frac{\sum_{n=51}^{150} s[n] * p_f[n]}{\sum_{n=51}^{150} p_f[n] * p_f[n]} \qquad \text{.....(6.5)}$$

The use of the central 100 sample section for this calculation reduces the influence of end of frame transitions. Amplitude changes in the input speech can thus be tracked more reliably.

The final prototype derived from the extraction process is thus:

$$p_{final}(n) = G_p * p(n), \quad n = 1,2,.....\tau \qquad .....(6.6)$$

An example of a set of gain adjusted prototypes, extracted from successive voiced speech frames of a female speaker, are shown in Figure 6.6.

6.2.3 Derivation of the 'Residual Prototype'

The final operation of the prototype extraction is to derive an LPC residual of the prototype. The residual is produced by filtering the prototype with the standard LPC filter (using the coefficients *a(k)* calculated for the current frame). It is, however, necessary to ensure that the residual reproduces the continuous nature of the extended prototype frame. This is achieved by ensuring that the history of the LPC filter contains a section of the end of the prototype prior to the filtering operation. The prototype residual is thus calculated over a prototype section of length $\tau + P_{LPC}$ and the history of the LPC filter is set to the last P_{LPC} samples of the prototype:

$$\begin{aligned} p_x(n) &= p_m(n+\tau-P_{LPC}) && \text{for } n = 1...P_{LPC} \\ p_x(n) &= p_m(n-P_{LPC}) && \text{for } n = P_{LPC}+1,..,\tau \end{aligned} \qquad .....(6.7)$$

$$e(n-P_{LPC}) = p_x(n) - \sum_{k=1}^{P_{LPC}} a(k) * p_x(n-k) \qquad .....(6.8)$$

$$\text{for } n = P_{LPC},..,\tau+P_{LPC}$$

The operation of equation (6.8) results in a prototype devoid of the formant structure determined by the LPC analysis of the current frame.

It is important to note that, following extraction, the residual prototypes are neither synchronous with the input speech or each other. Thus two prototypes extracted from successive speech frames would probably not join together smoothly. The extension of the prototypes to produce frame excitations would then cause significant discontinuities at frame boundaries. This would produce unacceptable audible distortion. It is thus necessary to align the residual prototypes of adjoining frames prior to reconstruction of the speech at the receiver.

6.3 Alignment of Residual Prototypes.

This section describes the technique employed to ensure that the prototypes selected, and used for reconstruction in successive frames, align optimally. This avoids impulsive auditory distortion in the output speech by allowing smooth interpolation between prototypes. The use of this technique removes the need to send details of the absolute position of prototypes with respect to the input speech. However, a disadvantage is that the input and synthesised speech waveforms are almost always non-synchronous, even in an unquantised coder.

The human auditory system is insensitive to the output speech phase shift since all frequencies of the input speech are equally affected, but, unfortunately, speech distortion measures (as discussed in chapter 3 under Objective testing) are not so tolerant. All of the Objective measures discussed (including the BSD discussed in chapter 5) perform measurements on a frame-by-frame basis and, since the frames of the input and synthesised speech will contain different and phase shifted waveforms, the measures are distorted. All correlation between human perception and the objective measures is thus lost.

The phase alignment technique presented here is an adaptation of that suggested by Kleijn for continuous prototypes [4]. The problem can be stated as:

Given two prototype residuals, $p_m(n)$ and $p_{m-1}(n)$ of lengths (pitch period) τ_m and τ_{m-1} ,respectively, time align the present prototype, $p_m(n)$, such that there is maximum cross-correlation between the two waveforms. This will allow smooth interpolation to be carried out between them.

The solution, described here, uses the discrete frequency domain as a convenient tool for manipulating prototypes of unequal length. The initial task is thus to calculate the DFTs of both the present and previous prototype residuals (with lengths τ_m and τ_{m-1} respectively). Note that the 'previous prototype', in this case, is the final τ_1 samples from the previous interpolated frame such that:

$$p_{m-1}(n) = e_f(L_f - \tau_1 + n) \quad for\ n = 1,\ldots,\tau_{m-1} \qquad \ldots\ldots\ldots\ldots(6.9)$$

where e_f is the previous frames interpolated excitation and L_f is the frame length.

The DFTs of the previous and present prototypes are denoted by $\mathbf{P}_{m-1}(k)\ for\ k = 1,\ldots,\tau_{m-1}$ and $\mathbf{P}_m(k)\ for\ k = 1,\ldots,\tau_m$ respectively. The interpolation process then consists of interpolation of the DFT coefficients. However, since τ_m and τ_{m-1} are not necessarily equal a method of interpolating between DFT series (and hence prototypes) of unequal length is required. This is achieved by adding zero 'harmonics' to the shorter prototype such that two prototypes of length τ are produced [4]. A further adjustment is made such that when τ_1 and τ_2 are related by a factor of 2 the shorter prototype is repeated such that both prototypes are of equal length. This caters for the phenomenon of pitch halving/doubling in speech whereby the fundamental frequency, as calculated by the pitch detector, alters by an integer multiplier, 2.

These adjustments produce two DFT series of adjusted length τ which we denote $\mathbf{P}'_{m-1}(k)$ and $\mathbf{P}'_m(k)$. The prototypes that these DFTs represent will, however, still be unaligned.

Smooth interpolation between prototypes requires that the prototypes be maximally aligned in the time domain. The alignment can be regarded as a rotation of $p_m(n)$ which is equivalent to a phase shift of $\mathbf{P}'(k)$ in the DFT domain. Kleijn, [4], suggests that the alignment should be performed on identically spectrally weighted prototypes. The weighting is similar to the spectral weighting in the CELP search process described in Chapter 3 It is important that the spectral weighting applied to the present and previous prototypes is identical and not affected by the spectral envelope changes of the input speech (characterised by the LPC coefficients $a(k)$). Both prototypes are thus weighted by a normalised filter based upon the present set of LPC coefficients. This ensures that the weighting operation is representative of the LPC inverse filtering operation and that the calculation can be performed at both transmitter and receiver. The spectral weighting operation is defined by:

$$\mathbf{A}(\mathrm{k}) = \sum_{\mathrm{n}=1}^{\mathrm{P_{LPC}}} \gamma^n a(n)\, e^{-j\left(\frac{2\pi kn}{\tau}\right)} \qquad \text{.....(6.10)}$$

$$\mathbf{W}(k) = \frac{\mathbf{A}(\mathrm{k})}{\mathbf{A}(\mathrm{k})\mathbf{A}^*(\mathrm{k})} \qquad \text{for } k = 0,1,\ldots,\tau$$

where γ is a weighting index, as used in the CELP search (normally ~0.8). The weighting de-emphasises the formants, since noise surrounding these is masked, while noise in areas of the spectrum away from formants will cause more auditory distortion.

The prototype DFTs are then weighted by $\mathbf{W}(k)$ to produce two weighted prototype DFTs:

$$\tilde{\mathbf{Q}}'_m(k) = \mathbf{P}'_m(k)\mathbf{W}(k)$$
$$\tilde{\mathbf{Q}}'_{m-1}(k) = \mathbf{P}'_{m-1}(k)\mathbf{W}(k) \qquad for\ k = 0,1,.....\tau \qquad .....(6.11)$$

Although this is a filtering operation performed by multiplication of DFTs it is not necessary to consider the problems of circular convolution since the results will not be inverse transformed to the time domain and exact equivalence is thus non-essential.

The time shift θ which maximise the cross correlation between the two prototypes can now be calculated as:

$$\theta = \underset{\theta'}{argmax} \sum_{k=0}^{\tau} \mathbf{Re}\left[\mathbf{Q}'_m(k)\mathbf{Q}'^{*}_{m-1}(k)\, e^{j\,2\pi k\,\theta'}\right] \qquad .....(6.12)$$
$$for\ \theta' = 0, 0.001, ..., 1$$

For convenience, θ is normalised to the pitch period. This simplifies the calculations of equations (6.12) and (6.13). The increment of θ by one thousandth was found by experimentation. Since this increment is non integer it actually causes further interpolation of the prototype by altering the phase of the DFT coefficients. This can be regarded as altering the sampling points of the original basis functions of the DFT and is thus equivalent to interpolating between the sampling points of the time domain prototype. The value of θ which maximises the value of the DFT cross correlation of equation (6.12) is then applied to phase shift the original aligned present prototype:

$$\tilde{\mathbf{P}}'_m(k) = \mathbf{P}'_m(k)\, e^{j\,2\pi k\,\theta} \qquad for\ k = 0,1,.....\tau \qquad .....(6.13)$$

6.4 Interpolation of Prototypes

The alignment process described in the previous section results in two aligned prototypes $\mathbf{P}'_{m-1}(k)$ and $\mathbf{P}'_{m}(k)$. These can now be smoothly interpolated. However, there is still a problem in that the prototype length must be extended/reduced over the interpolation interval. This results in the number of DFT coefficients describing the prototype varying over the interpolation interval and the key to the process is the introduction of a pitch counter C_p , which is defined as:

$$C_p = \sum_{i=0}^{p-1} \tau_i \quad for\ p = 0, 1, ... \qquad .....(6.14)$$

The upper limit in equation (6.14) is left undefined since this will be a function of the interpolation process. The definition of C_p in (6.14) describes the summation of all pitches τ_i from that of the previous prototype i.e. τ_0 to that of the previous prototype in the reconstructed excitation (τ_{p-1}).

It is now possible to define a linear interpolation coefficient, α, describing the level of contribution, in terms of coefficient magnitude and prototype length τ_i, to be taken from the new prototype. Similarly (1-α) will define the contribution level from the previous prototype.

Thus:

$$\alpha = \frac{C_p + \tau_p}{L_i}$$

where L_i is the interpolation interval in samples. .....(6.15)

α is constrained such that $\alpha <= 1$

It was found that the optimum interpolation interval L_i was half of the frame length i.e. 100. If the prototype extraction process is considered, it is reasonable that, since the prototypes are chosen only from the current frame, the current prototype should be reached at a point central in that

frame. Other authors [5] have suggested that the new prototype should be reached at the end of the interpolated frame. The latter was found to overdamp the reaction of the coder to amplitude changes especially at the beginning of words.

The interpolation algorithm then proceeds by defining the prototype length (pitch) to be used for each of the prototypes constituting the interpolation interval:

$$\tau_p = (1-\alpha)\tau_{m-1} + \alpha\tau_m \qquad \text{.....(6.16)}$$

The interpolated excitation is then defined as:

$$e_i(t+C_p) = \mathbf{Re} \sum_{k=1}^{\tau_p} \left((1-\alpha)\mathbf{P}_{m-1}(k) + \alpha\,\mathbf{P}_m(k)\right) e^{j\frac{2\pi kt}{\tau}} \qquad \text{.....(6.17)}$$

$$for\ t = 0,1,\ldots..,\tau_p$$

Throughout the process the conjugate symmetry of the DFT of a real sequence is maintained even though the number of points of each effective IDFT varies. The sequence of equations (6.14-6.17) define the interpolation process and are repeated until $(t+C_p)$ exceeds the interpolation frame length L_f (in this case 200 samples).

The final, interpolated prototype excitation $e_i(t)$ is then filtered by the IIR LPC inverse filter to produce a reconstructed speech frame. The nature of this filter results in smoothing of any discontinuities generated by the interpolation process. In practice the evolution of the prototypes across the frame excitation has been found to produce few discontinuities. An example of the processing of a number of voiced speech frames, using the prototype technique presented here, is shown in Figure 6.7. The prototypes and the resulting interpolated excitation are unquantised. It can be seen that the interpolated prototype excitation approximates the original excitation well and that the output speech follows the behaviour

of the input closely. As expected, the output speech and its corresponding excitation are not synchronous with the input and its residual. The amplitude and pitch behaviour are, however, closely modelled.

6.5 Quantisation of Prototypes

Two quantisation techniques have been implemented; one operates in the DFT domain and the other in the time domain. The major challenge presented by the quantisation is the variable length of the prototypes (from 16 to 147 samples). This makes the quantisation procedure more complex than a simple CELP scheme where a fixed sub-frame length is used. In the time domain, however, impulsive codebooks, with variable length vectors, are successfully employed. In the DFT domain a scheme which differentially codes a limited set of coefficients is discussed.

6.5.1 DFT Coefficient Quantiser

In section (6.4) of this chapter a technique for prototype extraction was discussed. This derives an unaligned prototype representation for each frame with length τ_m. For quantisation purposes the prototype can be regarded as a time series, $p(n)$, which has an equivalent series of τ_m DFT coefficients $\mathbf{P}(k)$. For 8kHz sampled speech the coefficients represent frequencies up to the Nyquist frequency of 4kHz, however previous work [6] on sub-band coders and the LPC excitation suggest that the excitation energy beyond 1kHz is severely reduced. Thus, the coefficients of major significance can be regarded as those below 1kHz. This premise is confirmed by the previous work of chapter (4) which showed the spread of 'essential' coefficients in the LPC excitation of a CELP architecture.

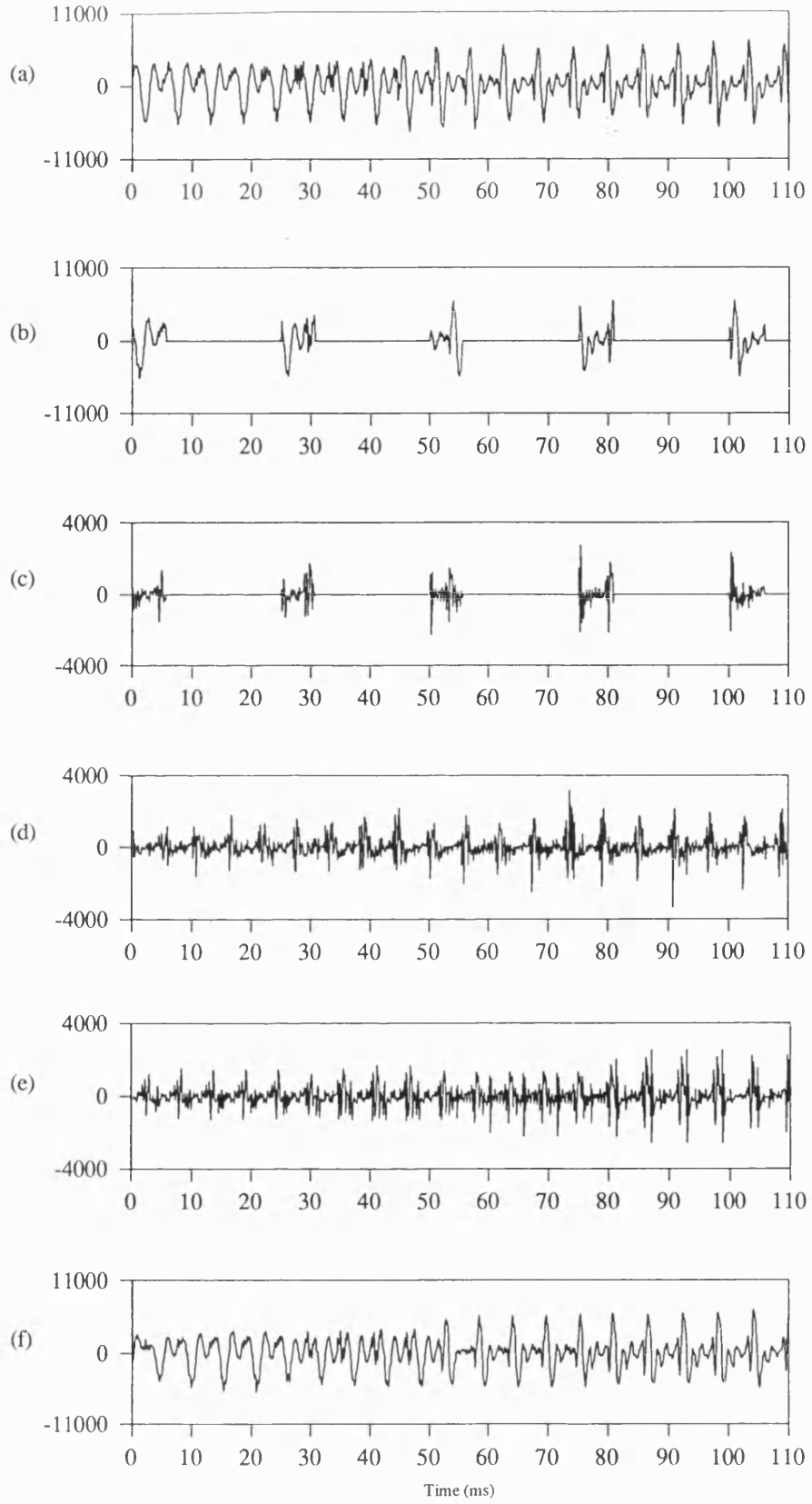


Figure 6.7: The operation of the prototype waveform coder: (a) The input speech. (b) Extracted prototypes. (c) Residual of prototypes. (d) Reconstructed Interpolated excitation. (e) Actual speech residual. (f) Reconstituted output speech.

For coding purposes, the first four coefficients (excluding the d.c. coefficient) are quantised using a differential scheme which is discussed later. Since the prototypes are one pitch period long these coefficients correspond to the fundamental and the first, second and third harmonics. The quantisation of these coefficients, is thus related to the harmonic and sine-wave speech coders described in [7][8].

In practice, it is also necessary to provide some representation for frequencies above 1kHz. This should avoid the piped speech phenomena which is a characteristic of low rate coders. Gupta and Atal [9] suggest a scheme for deriving bandwidth enhanced coefficients to supplement the sub-1kHz representation.

The effective bandwidth corresponding to a DFT coefficient (i.e. the 4dB bandwidth of each related bandpass filter [10]) representing a series of N coefficients is defined by:

$$BW_c = \frac{f_s}{N} \text{ where } f_s \text{ is the sampling frequency (8KHz) .....(6.18)}$$

Thus, taking an example pitch period of 40 samples, the effective coefficient bandwidth is 200 Hz. If we consider the speech to be band

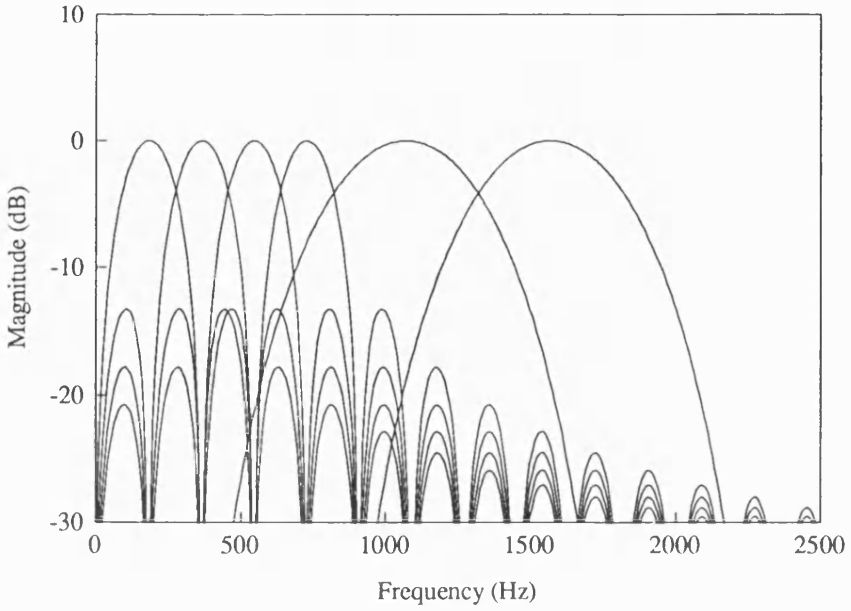


Figure 6.8: Coefficient and enhanced coefficient filter bank for a prototype length of 44 samples.

limited to 3.4kHz and we assume that two coefficients can be quantised the bandwidth must be extended to some 1200 Hz. However, since the excitation has low spectral energy at high frequencies this can be reduced to ~500Hz. This represents the spectrum from 1kHz to 2 kHz with two quantised coefficients.

The bandwidth expansion procedure [9] is performed by using a reduced width, windowed basis function (If the DFT is considered as a correlation procedure, then the exponential series, to be correlated with the input series, is the basis function). A Hamming window of length 25 samples was used, giving an expanded coefficient bandwidth of 320 Hz. When the coefficients are carefully positioned this gives adequate spectral coverage. A smaller window length would widen the coefficient bandwidth, however this severely limits the number of samples in the input sequence represented by the high-order basis functions. For instance, a window length of 16 would represent just 20% of an 80 sample prototype.

Long prototypes, as produced by speakers with low pitch, also cause a problem if fixed filters are to be employed. A prototype length of 147 samples has ~18 coefficients below 1kHz. In a low rate coding scheme only a limited number of these will be quantised. Thus, for long prototypes, the high order filters must also provide information for the sub-1kHz spectrum. The filters are thus made mobile with respect to the last quantised low order coefficient.

For the bandwidth expansion, The Hamming window is applied to the Fourier basis function such that:

$$f_k^w(n) = e^{-j\frac{2\pi k n}{\tau_m}}\left[0.54 - 0.46\cos\left(\frac{2\pi n}{\tau_m}\right)\right] \qquad \text{.....(6.19)}$$

$$for\ n = 0, 1, \ldots\ldots \tau_{m-1}$$

Gupta, [9], notes that these functions will not necessarily be orthogonal and that optimal coefficients can be calculated by Least Squares techniques. For simplicity, however, a degree of orthogonality is assumed. This allows the bandwidth enhanced coefficients to be calculated as:

$$C^w(k) = \frac{1}{\tau_m} \sum_{n=0}^{\tau_m - 1} x(n) f_k^w(n) \qquad \text{.....(6.20)}$$

Although direct quantisation of the coefficients is possible it is unlikely to give good results when so few bits are available for quantisation (for a sub-3kbit/s coder approximately 36 bits can be used for excitation quantisation). A differential scheme was thus implemented whereby the quantised difference between the previous quantised coefficients and those derived from the present prototype are transmitted. For these purposes each prototype is aligned with its peak amplitude as the central sample. This minimises overall phase variation between successive prototypes. An alternative scheme uses the weighted alignment procedure described in section (6.3).

The first four (low order) DFT coefficients are then derived using the standard DFT calculation and the high order enhanced coefficients using equation (6.20). These coefficients are positioned at a 500Hz interval with the first coefficient 250Hz above the fourth low order DFT filter's upper 4dB frequency.

The coefficients are quantised by searching a codebook of 8 differential increments for each real and imaginary coefficient element. The quantisation operation is described by:

$$\mathbf{Re}(\mathbf{Q}_m(k)) = \underset{q'}{\operatorname{argmin}} \left[\mathbf{Re}(\mathbf{P}_m(k)) - \mathbf{Re}(\mathbf{Q}_{m-1}(k)) - \mathbf{C}(q') \right] \text{.....(6.21)}$$

where $\mathbf{P}_m(k)$ is the current calculated coefficient and $\mathbf{Q}_{m-1}(k)$ is the previously quantised coefficient. A similar expression is used for the

imaginary part of the coefficients. The codebook $\mathbf{C}(q')$ was found by experimentation and has values:

$$\mathbf{C}(q') = \{ -100, -20, -5, -1, 1, 5, 20, 100 \}$$

Improvements could be made by training this codebook if sufficient speech data were available. Since an 8 level coder is used for each real/imaginary part of the complex coefficients and four original and two enhanced coefficients are coded, a total of 36 bits are required to code the prototype excitation.

The DFT coefficient quantisation procedure was used to quantise prototypes in the prototype coder operating on the Bath speech database. The technique produces intelligible speech for most speakers, but has a hollow quality. This can be annoying for listeners and is due to the poor spectral representation of the prototypes for some speakers. This is particularly true for long pitch period speakers i.e. low pitched male speakers where few high frequency components are represented. The operation of the coefficient quantiser is shown in Figure 6.9. Waveform (e), which shows the quantised and interpolated prototype excitation, illustrates the low pass effect of the scheme.

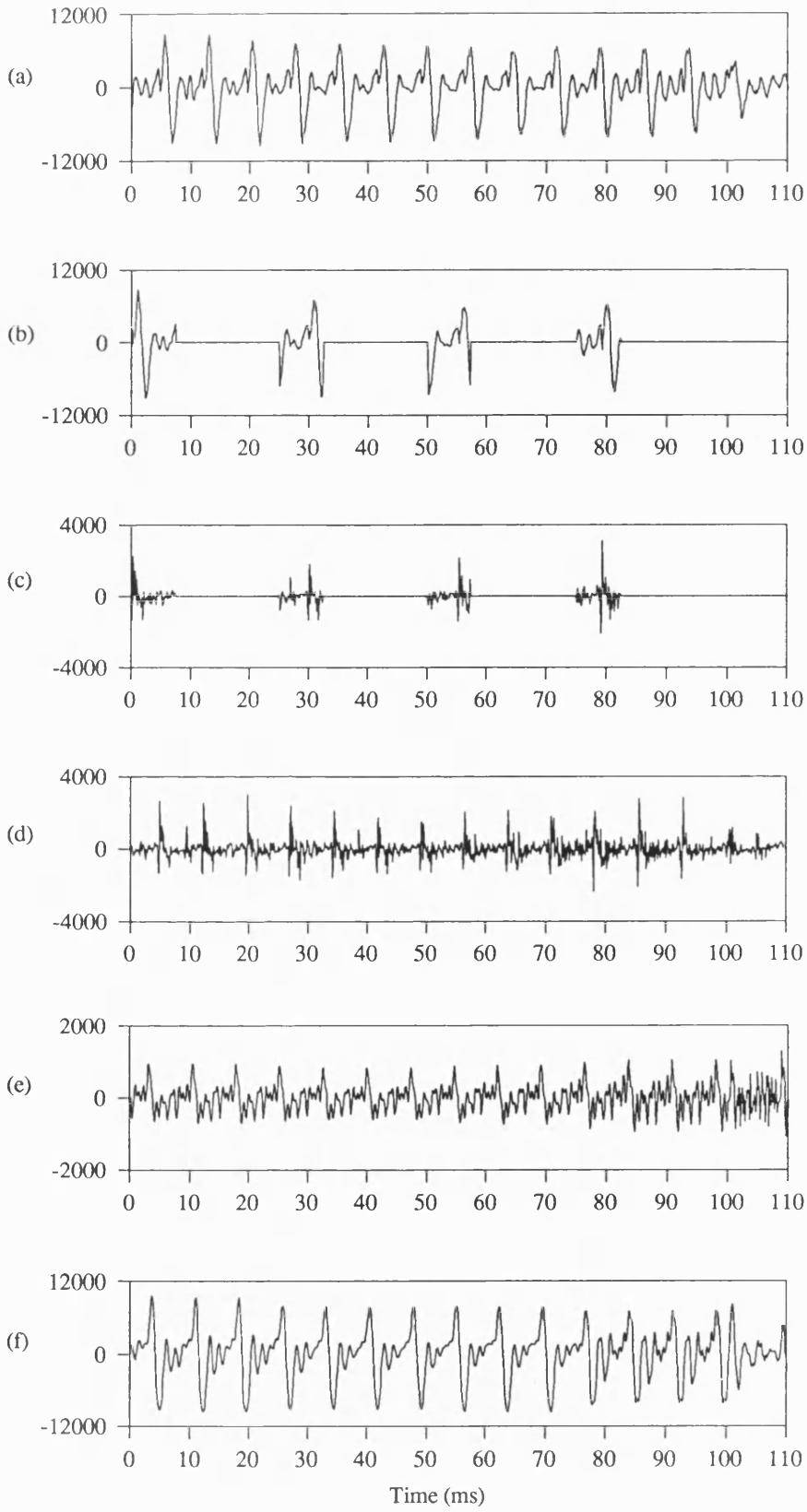


Figure 6.9: Prototype Waveform Coder using DFT Coefficient Quantiser: (a) The input speech. (b) Extracted prototypes. (c) Residual of prototypes. (d) Reconstructed Interpolated Excitation. (e) Actual speech residual. (f) Reconstructed output speech.

6.5.2 Impulsive Quantiser

This section considers quantisation of the prototypes in the time domain. This is similar to the quantisation technique described by Granzow et. al. [11][12]. The quantisation consists of the searching of two codebooks populated by impulses and the calculation of a differential gain component from the previously quantised prototype. The scheme is, thus, similar to the CELP schemes discussed in previous chapters.

The quantisation procedure can be summarised by:

$$\mathbf{u_m}(n) = \alpha_0 \mathbf{v_0}(n) + \alpha_1 \mathbf{v_1}(n) + \beta\, \mathbf{u_{m-1}}(n) \qquad for\ n = 0,1,.....\tau_m \qquad .....(6.22)$$

where $\mathbf{v}_0(n)$ and $\mathbf{v}_1(n)$ two codebook vectors and $\mathbf{u}_m(n)$ and $\mathbf{u}_{m-1}(n)$ are the new and previously quantised prototype, respectively. The gain terms α_0, α_1 and β are similar to the gain terms calculated in the CELP search (see 3.4.3). The first codebook vector $\mathbf{v}_0(n)$ is derived from a codebook of single delta impulses of unit amplitude. Thus for a given pitch delay τ_m:

$$\mathbf{v_o^k}(n) = \begin{cases} 0 & for\ n \neq k \\ 1 & for\ n = k \end{cases} \qquad for\ n = 0,1,...\tau_m \qquad .....(6.23)$$

where **k** is the codebook index. There are, effectively, 128 possible vectors in this codebook. As an excitation this codebook will produce output speech of the form of the inverse LPC filter impulse response starting at a point k.

The second codebook vector $\mathbf{v}_1(n)$ is derived from a 128 vector ternary codebook, formed by centre clipping a gaussian codebook. The codebook used was that defined in the US Federal Std. 1016 4.8kbit/s speech coder [13] which clips the codebook values at ±1.2.

Although the searching of these codebooks is comparable to a CELP search algorithm there is an important difference. These searches are performed between two like vectors (i.e. the LPC excitation) whereas the

CELP search is performed with the inclusion of the LPC synthesis filter. The CELP search can, thus, choose an LPC excitation vector to optimally synthesise the input speech sub-frame, while prototype codebooks code the excitation in an open-loop manner.

Kleijn [4] suggests a weighting scheme for prototype codebook searches using a perceptual error weighting filter applied to both vectors. This acts as a weighted distortion measure. As in CELP the weighting filter is derived from the LPC synthesis filter, but for simplicity, the impulse response is truncated to 25 samples. This was noted as being acceptable in section (4.1.4). The perceptual weighting filter is derived by applying the weighting factor γ to the impulse response such that:

$$\mathbf{h}=\left\{h(0)+\gamma\, h(1)+\gamma^2 h(2)+.....\gamma^{24}h(24)\right\} \qquad .....(6.24)$$

The weighting operation can then, conveniently, be expressed in matrix form as:

$$\mathbf{y}=\mathbf{Hu} \qquad .....(6.25)$$

where the weighing matrix **H** is defined as:

$$\mathbf{H}=\begin{vmatrix} h(0) & \ldots & \ldots & \ldots & \ldots & 0 \\ \gamma\, h(1) & h(0) & \ldots & \ldots & \ldots & \ldots \\ \gamma^2 h(2) & \gamma\, h(1) & h(0) & .. & \ldots & \ldots \\ \gamma^3 h(3) & \gamma^2 h(2) & \gamma\, h(1) & \ldots & \ldots & \ldots \\ .... & \gamma^3 h(3) & \gamma^2 h(2) & \ldots & \ldots & \ldots \\ \ldots & \ldots & \gamma^3 h(3) & \ldots & \ldots & h(0) \\ \gamma^{25} h(25) & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \gamma^{25} h(25) & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \gamma^{25} h(25) & \ldots & \ldots & \ldots \\ 0 & \ldots & \ldots & \ldots & .0 & \gamma^{25} h(25) \end{vmatrix}$$

.....(6.26)

The two codebook searches, using the weighting operation, are performed in a necessarily sub-optimum sequential fashion. Prior to these

operations, however, the optimum contribution from the previous prototype (i.e.β) must be calculated. If the prototype were taken as extracted they would be unaligned, making the possibility of a meaningful differential contribution poor. The alignment operation of (6.25) is thus performed such that the prototypes are positioned for maximum time alignment making a meaningful calculation of β possible.

The value of β is calculated, in a similar way to the CELP gains, as:

$$\beta = \frac{\sum_{n=0}^{\tau_m - 1} p'_m(n).p_{m-1}(n)}{\sum_{n=0}^{\tau_m - 1} p_{m-1}(n).p_{m-1}(n)} \qquad \text{.....(6.27)}$$

The contribution of the previous prototype is then subtracted from the prototype such that:

$$p'_m(n) = p'_m(n) - \beta\, p_{m-1}(n) \quad \text{for } n = 0,1,.....\tau_m - 1 \text{.....(6.28)}$$

The two impulsive codebook searches are then performed so as to minimise the squared weighted error between the candidate vector and the prototype remainder. This is described in matrix terms by:

$$k = \underset{k'}{\text{argmin}} \; (\mathbf{e} - \alpha\, \mathbf{v}_{k'})^{\mathrm{T}} \mathbf{H}^{\mathrm{T}} \mathbf{H} (\mathbf{e} - \alpha\, \mathbf{v}_{k'}) \qquad \text{.....(6.29)}$$

In a, perhaps, more meaningful non matrix form this can be expressed as:

$$k = \underset{k'}{\text{argmin}} \sum_{n=0}^{\tau_m - 1} \left(\left[e(n) - \alpha\, v_{k'} \right] * h(n) \right)^2 \qquad \text{.....(6.30)}$$

where $e(n)$ is the adjusted value of $p'_m(n)$ for the first codebook search.

For the second codebook search:

$$e(n) = p'_m(n) - \alpha_0 v_{k'}(n) \quad \text{for } n = 0,1,.....,\tau_m - 1 \qquad \text{.....(6.31)}$$

For both codebooks the value of α is calculated for each vector prior to the evaluation of equation (6.29) according to:

$$\alpha_{0 \text{ or } 1} = \frac{\sum_{n=0}^{\tau_m - 1} u_h(n).v_h(n)}{\sum_{n=0}^{\tau_m - 1} v_h(n).v_h(n)} \qquad \text{.....(6.32)}$$

where $u_h(n) = u(n) * h(n), \quad v_h(n) = v(n) * h(n)$

The complete search procedure can be summarised as:

- Align present and previous quantised prototype.
- Calculate β and subtract contribution from prototype. (equations 6.27 and 6.28)
- Search codebook 1 for impulse contribution and gain α_0. (equations 6.30 and 6.32)
- Use equation 6.31 to calculate *u(n)*.
- Search codebook 2 for ternary contribution and gain α_1.

Following the searches, the gains α_0 ,α_1 and β are quantised using 5 bits. The quantisation levels were determined from a speaker sample of 20 male/female speakers. The distributions of these gains are shown in Figure 6.10.

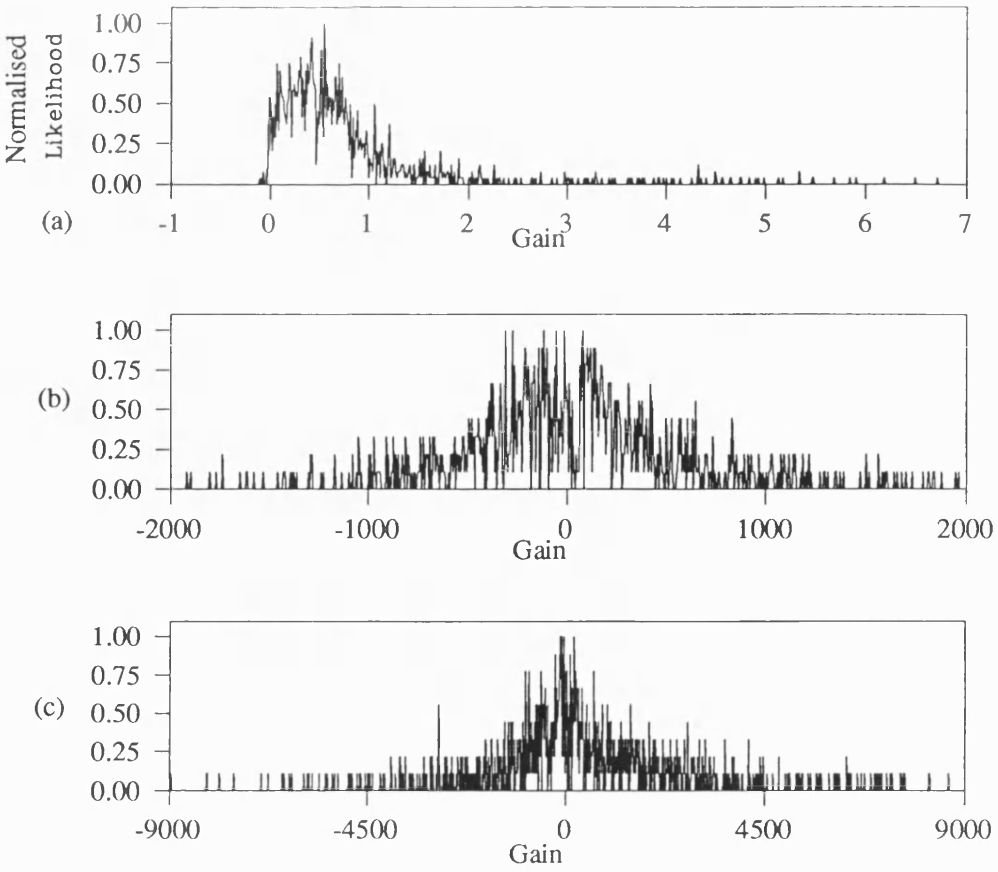


Figure 6.10: Gain distributions for Impulsive Codebook Quantiser. (a) ,the gain β of previous prototype contribution and (b), (c) the gains α_0 ,α_1 for the single and multi-impulse codebooks, respectively.

The distribution for β is almost entirely positive due to the time alignment operation performed prior to the gain calculation. The distributions of α_0 and α_1 are similar to those found for CELP gain terms. The overall bit rate required to code the prototype excitation using this technique is (3*5) + (2*7)=29 bits if 128 vector codebooks are used. Figure 6.11 shows sample waveforms from the quantisation of a prototype using this procedure. For comparison, waveforms from the coefficient coder have also been included. In general, the impulsive quantiser results in clearer, but harsher, speech than the coefficient quantising scheme described in the previous section.

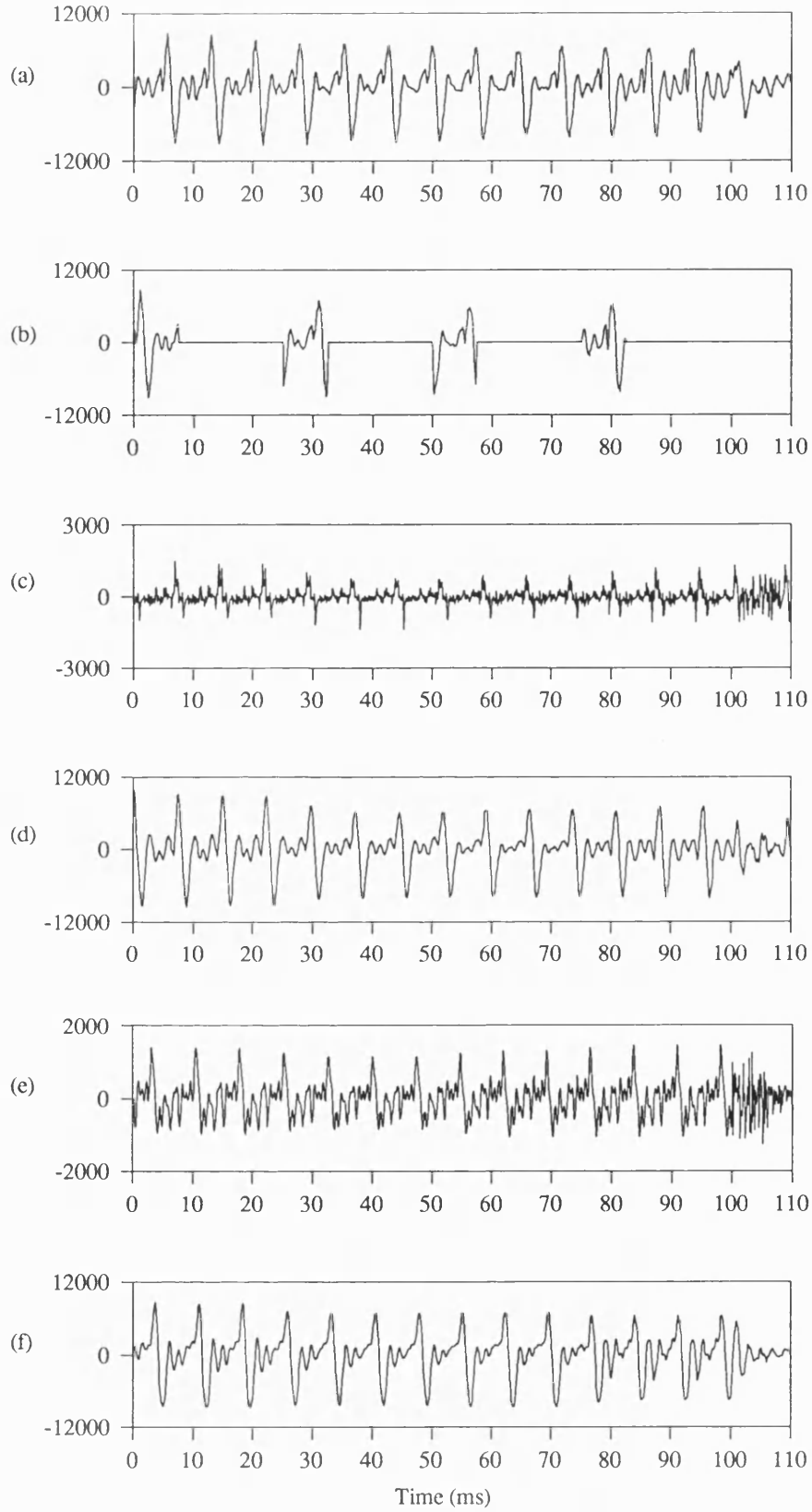


Figure 6.11: Comparison of output of Impulsive and Coefficient quantisers: (a) Input speech. (b) Extracted prototypes. (c) & (d) Excitation and output speech for impulsive quantiser. (e) & (f) Excitation and output speech for coefficient quantiser.

6.6 Unvoiced Frame Coding

Frames declared unvoiced by the pitch determination algorithm described in section (6.1) are coded using a CELP algorithm. Since unvoiced frames do not have a significant periodic component the Long Term Predictor (LTP) is excluded, leaving just the fixed codebook search. Some authors [14] suggest the retention of the LTP for unvoiced frames. The intention of the latter being that voiced frames mistakenly classed as unvoiced can still be coded. This is, however, a waste of bits for a low rate coder. The pitch determination algorithm discussed previously biases decisions in favour of voiced frames, thus avoiding the problem.

Unvoiced sub-frames are coded using a 128 vector overlapped gaussian codebook. Each 200 sample frame is divided into three sub-frames of length 67, 67 and 66 samples. The gains for each sub-frame are coded with 5 bits according to the US-Federal. Std. 1016 scheme. The total bit rate for coding of an unvoiced frame excitation is thus 3 x 7 + 3 x 5 = 36 bits. This is comparable with the total bits required for the coding of voiced frames using the impulsive codebook scheme. The codebook searches are performed using the standard perceptually weighted squared error measure.

The output excitation vectors from both the voiced and unvoiced frame coders are inverse filtered by the common LPC inverse filter. This effectively smoothes any discontinuities at unvoiced/voiced frame borders. The smooth transition is further enhanced by taking the following voiced frame's previous excitation to be the last τ_m samples of the unvoiced frames excitation.

6.7 Combination of Prototype and CELP algorithms

The combined algorithms are forthwith referred to as a mixed prototype waveform / CELP scheme (PW/CELP). Figure 6.12 shows the basic architecture of the PW/CELP coder. In terms of coding the two constituent coders operate independently for voiced and unvoiced frames respectively. There is, however, a common requirement for the derivation and quantisation of the 10 LPC coefficients.

The LPC coefficients are calculated using a windowed autocorrelation analysis based on the Levinson recursion (see Chapter 3). The coefficients are then quantised by use of Line Spectral Frequencies. The LSFs can be calculated using either of the algorithms discussed in section (3.2.3), however the method of [15] is clearly preferable due to its efficiency. The LSFs are currently quantised using the US Federal Std 1016 4.8kbit/s scheme, which requires 34 bits per 200 sample frame. Combined with the prototype quantisation requirement of 37 bits this gives a minimum bit rate for the coder of 2480 b/s. However, Paliwal and Atal [16] have shown that a split vector quantisation scheme (split-VQ) can encode the LSFs in just 25 bits. (They also describe a scheme using 24 bits with increased distortion.).

The split-VQ algorithm codes the LSFs as two vectors. The first vector consists of the first four LSFs and the second, the remaining six. Twelve bits are then allocated to each part giving a minimum total of 24 bits. The quantisation codebooks are searched using a weighted distance measure which weights the quantisation of the lower LSFs compared to the higher LSFs. This corresponds to the weighting of the human ear discussed in chapter 2.

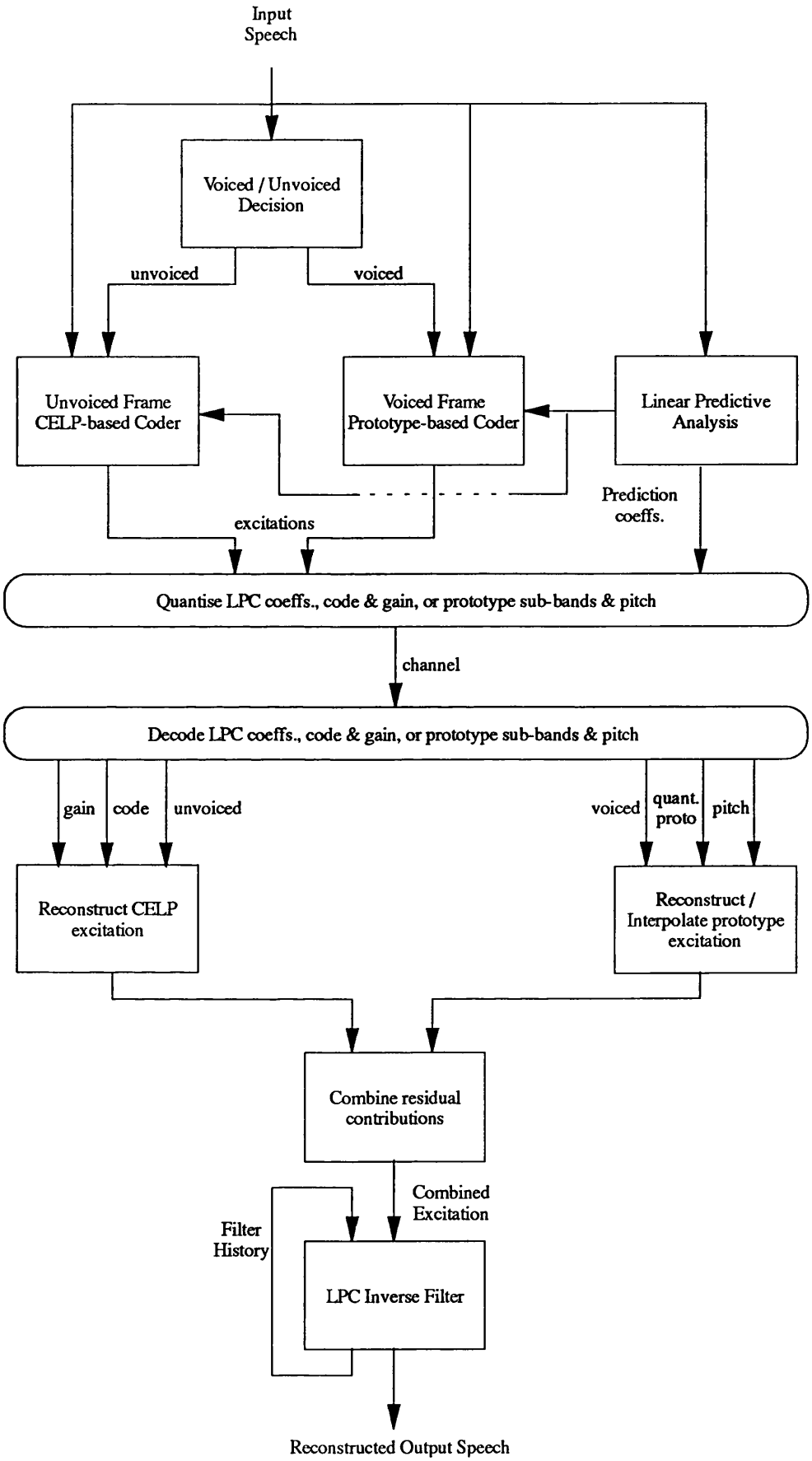


Figure 6.12: The mixed Prototype Waveform CELP algorithm (PW/CELP).

It has not been possible to use the split-VQ scheme owing to the requirement for a large speech training base. Paliwal and Atal used a database of FM radio recordings amounting to 20+ minutes of 170 speakers. This gives 60000 LSF vectors for training. The current Bath database of 5 minutes of speech from 30 speakers is thus vastly inferior and not suitable for such a training exercise. However, since Paliwal and Atal show that with 25 bits the spectral distortion is <1dB it is valid to use the US Federal STD coder for comparative tests. In practice the latter is likely to have increased distortion compared to the split-VQ scheme.

Voiced Frame:	**Prototype Waveform Coder**			
LSF Quantiser:	**US Def Std**	**Split VQ**	**US Def Std**	**Split VQ**
LSFs:	34	25	34	25
Pitch	7	7	7	7
Voiced\Unvoiced	1	1	1	1
Proto. Quantiser:	**Impulsive Codebook**		**DFT Coefficient**	
Prototype Coding:	29	29	36	36
V. Frame Bits:	**71**	**62**	**78**	**69**
Unvoiced Frame:	**Single Codebook CELP Coder**			
LSFs:	34	25	34	25
Gain Terms (3x)	15	15	15	15
Codebook Index (3x)	21	21	21	21
UV. Frame Bits:	**70**	**61**	**70**	**61**
Max. Bits /Frame	**71**	**62**	**78**	**69**
Overall Bit Rate	**2840 b/s**	**2480 b/s**	**3120 b/s**	**2760 b/s**

Table 6.1: Bit Allocation between parameters for Voiced and Unvoiced frames in PW/CELP. Bit rates for both prototype and LSF quantiser are shown.

6.7.3 Bit Allocation for PW/CELP

The bit allocation for voiced and unvoiced schemes using the various parameter quantisation schemes is described in Table 6.1. From the bit rates, it is clear that the PW/CELP scheme can produce good quality speech at bit rates considerably lower than CELP. Most of the advantage

comes from the removal of the Long Term Predictor and more efficient coding of voiced frames.

6.7.4 Results of PW/CELP

Coded speech from the PW/CELP algorithm (using the impulsive codebook prototype quantisation) is shown in Figure 6.13. The synthesised speech is compared with the original for unvoiced, voiced and transitional sections. The waveforms show that the coder generally reproduces voiced speech well, with the basic shape evolving similarly to the input speech. However in transitional unvoiced/voiced section the interpolation procedures slow the coders response and the synthesised speech fails to reproduce the fast waveform changes of the input. These phenomena clearly degrade the synthesised speech quality but appear unavoidable when a long frame is chosen so as to reduce overall bit rate.

Since objective measures are not applicable to the non-synchronous PW/CELP coder informal listening tests have been performed. These show that PW/CELP is currently between low rate LPC-10 vocoders and the higher rate CELP algorithms in performance. The tonal artefacts produced by the DFT coefficient quantisation technique significantly degrade listening quality. PW/CELP, using the impulsive quantiser, produces considerably more natural sounding speech than low rate vocoders. The clarity of the 4.8kbit/s CELP algorithms is, however, missing.

In considering these results it should be remembered that the 4.8kbit/s coding algorithm is a highly developed coder. Prototype waveform techniques have only recently been introduced and already show promising results at rates lower than standard CELP techniques.

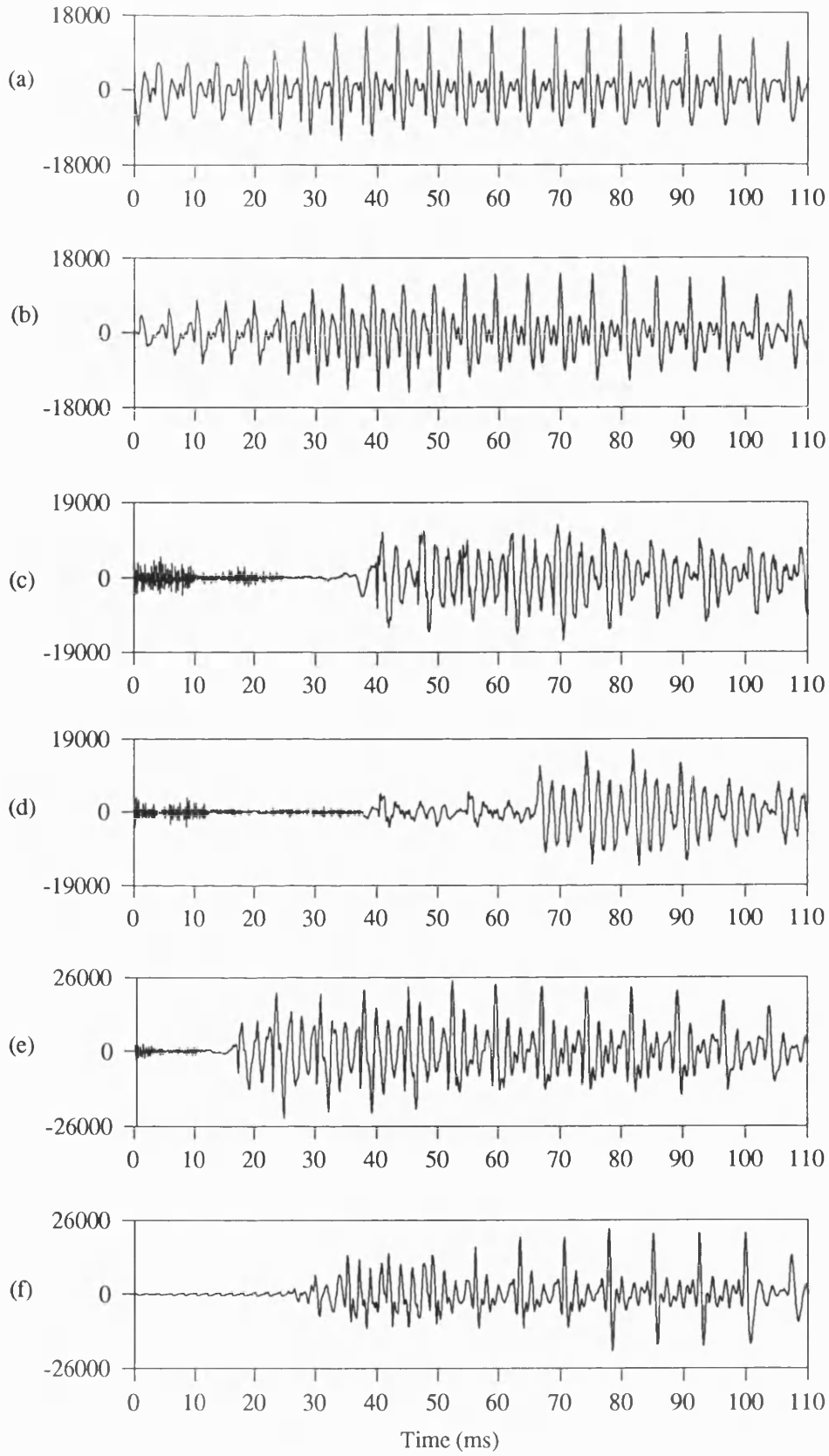


Figure 6.13: Waveforms from a fully quantised PW/CELP implementation at 2.84kbit/s. (a), (c),and (e) show input speech sections and (b), (d) and (f) the respective synthesised sections.

6.8 Conclusions

Current low-rate speech coders do not differentiate between voiced and unvoiced frames, but further bit-rate reductions require such a switch to be re-introduced. This reduces the information requiring transmission in both frame types, and the prototype waveform technique was introduced as a new technique for coding voiced frames. Each voiced frame is represented by a single pitch-period prototype, which is selected using a new MSE criterion, computed between a frame of repeated prototypes and the input speech. Following prototype selection, a residual prototype is derived using standard LPC analysis.

Residual prototypes are interpolated between frames using a new interpolation technique. This operates in the DFT domain and interpolates both the shape and pitch period of the prototypes. Such a technique produces a smooth evolution between successive speech frames. The standard LPC synthesis filter is used to synthesise speech from the interpolated excitation and, while good quality synthesised speech is produced, the technique does not preserve the phase of the input. Objective measures of coder performance were, thus, found to be unsuitable for the evaluation of prototype waveform coding.

Two residual prototype quantisation schemes were presented. The first, a DFT coefficient coder, codes the prototype pitch harmonics, concentrating on the low order coefficients. Higher frequency bands were represented by the use of 'Bandwidth-enhanced' coefficients. Such a scheme, produces intelligible synthesised speech but informal listening tests suggest that the accompanying hollowness is perceptually annoying. A second scheme, an Impulsive quantiser, codes the prototype in the time-domain as a combination of two gain adjusted codebook vectors and a differential component. This scheme generates higher quality speech and can be coded at a lower bit-rate than the DFT coefficient scheme.

The Prototype waveform coder was combined with CELP, for the coding of unvoiced frames, to produce a fully quantised PW/CELP coder, operating at less than 3kbit/s. This coder produces acceptable speech which, though not toll-quality is significantly more natural than low rate vocoders. This naturalness contrasts with the robotic quality of previous techniques and is due to the enhanced reproduction of pitch periodicity. One significant problem with the technique, however, is its failure to track fast amplitude changes in the speech waveform; this reduces coded speech intelligibility

6.9 References

[1] L. R. Rabiner and R. W. Schafer, "Digital Processing of Speech Signals," *Prentice-Hall Signal Processing Series*, 1978.

[2] J. J. Dubnowski, R. W. Schafer and L. R. Rabiner, "Real-Time Digital Hardware Pitch Detector," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. 24, No. 1, pp. 2-9, Feb. 1976.

[3] R. E. Crochiere and L. R. Rabiner, "Multirate Digital Signal Processing," *Prentice-Hall Signal Processing Series*, 1983.

[4] W. B. Kleijn, "Continuous Representations in Linear Predictive Coding," *Proc. Int. Conf. on Acoust., Speech and Signal Proc.*, pp. 201-204, May 1991.

[5] J. Haagen, H. Nielsen and S. D. Hansen, "Improvements in 2.4Kbps High-Quality Speech Coding," *Proc. Int. Conf. Acoust., Speech and Signal Proc.*, pp. 145-148, March 1992.

[6] N. G. Kingsbury, "Robust 8000 bit/s sub-band speech coder," *IEE Proceedings*, Vol. 134, Pt. F, No. 4, pp. 352-366, July 1987.

[7] I. M. Trancoso, L. B. Almeida and J. M. Tribolet, "A Study on the Relationships between Stochastic and Harmonic Coding," Proc. Int. Conf. Acoust., Speech and Signal Proc., pp. 1709-1712, Tokyo, 1986.

[8] R. J. McAulay and T. F. Quatieri, "Sine-Wave Phase Coding at Low Data Rates ," *Proc. Int. Conf. Acoust., Speech and Signal Proc.*, pp. 577-580, May 1991.

[9] S. K. Gupta and B. S. Atal, "Efficient Frequency-Domain Representation of LPC-Excitation," *IEEE Workshop on Speech Coding for Telecomms.: Digital Voice for the Nineties*, pp. 64-65, Sept. 1991.

[10] E. O. Brigham, "The Fast Fourier Transform and its Applications," *Prentice-Hall Signal Processing Series*, (Chapter 13), 1988.

[11] W. Granzow and B. S. Atal, "High-Quality Digital Speech at 4KB/S," *Proc. IEEE Global Telecomms. Conf.* , pp. 941-945, 1990.

[12] W. Granzow, B. S. Atal, K. K. Paliwal and J. Schroeter, "Speech Coding at 4KB/S and Lower Using Single-Pulse and Stochastic Models of LPC Excitation," *Proc. Int. Conf. Acoust., Speech and Signal Proc.*, pp. 217-220, May 1991.

[13] U.S. National Communications System, Washington, D.C., "Proposed Federal Standard 1016, Second Draft," Nov. 1989.

[14] K. Y. Lo, B. M. G. Cheetham, W. T. K. Wong and I. Boyd, "A Pitch Synchronous Scheme for Very Low Bit Rate Speech Coding," *IEE Colloquium on "Speech Coding - Techniques and Applications"*, pp. 3/1-3/5, 14th April 1992.

[15] P. Kabal and R. P. Ramachandran, "The Computation of Line Spectral Frequencies Using Chebyshev Polynomials," *IEEE Trans. on Acoust., Speech, and Signal Proc.*, Vol. 34, No. 6, pp. 1419-1426, Dec. 1986.

[16] K. K. Paliwal and B. S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame," *Proc. Int. Conf. Acoust., Speech and Signal Proc.*, pp. 661-664, May 1991.

Chapter 7: Conclusions and Further Work

This thesis has considered the use of hybrid time-frequency domain techniques for the coding of speech at low/medium bit rates. The coders developed, have shown that improvements in both coded speech quality and reductions in transmission bit-rates are produced by combining operations in both domains. This chapter reviews the techniques and suggests areas of further work.

7.1 Frequency Domain searched CELP

The initial investigations into Frequency Domain CELP (Code Excited Linear Prediction) architectures provided the basis for the more advanced coders considered in later sections of the thesis. These investigations, showed that by holding and searching a fixed codebook in the Discrete frequency domain, computational complexity is reduced. Further, when combined with the new Overlapped Frequency domain codebook, the Frequency domain CELP architecture is comparable in both complexity and storage terms with Time-Domain Overlapped Codebook techniques. The new Overlapped Frequency Domain CELP produces speech with a SEGSNR within 0.5dB and AV.SNR within 0.1dB of standard Time Domain CELP performance.

While the move to a transform domain does not offer any reduction in bit-rate or direct improvement in speech quality, it does allow analysis of the 'pseudo-ideal' excitation of the CELP architecture. This was exploited to reveal a number of features of CELP and LPC (Linear Predictive Coding), in particular:

- A few 'essential' frequency components (e.g. 5) used as an excitation can produce high quality speech.
- The adaptive codebook/LTP pitch predictor fails to fully represent the low frequency content of the speech.

The fact that an excitation, consisting of a limited number of 'essential', and perceptually significant, coefficients, can produce high quality speech suggests that a higher degree of perceptual information should be employed in the CELP search. Further, if the pitch periodicity of speech is to be adequately represented in lower bit-rate coders, an alternative approach to coding the speech pitch component is required. Chapters 5 and 6 considered new coding structures based on these conclusions.

7.2 Improvements in Perceived Speech Quality

In Chapter 5, the Bark Spectral Distortion (BSD) was integrated into both Time and Frequency domain searched CELP. The BSD was developed as a model of the psycho-acoustic and physiological processes of the human ear and improves on the simple perceptual weighting employed in standard CELP. It was shown that BSD searched CELP improves the perceived quality of the coded speech, while using a standard CELP codebook structure and hence requiring no increase in bit-rate. In practice, if complexity allows, the BSD search can be used as a direct replacement for the MSE technique in current CELP coders. Further quality improvements were produced by increasing the density of the Critical Band filter functions used in the BSD; while again increasing complexity, this makes the BSD approximate the behaviour of the human ear more closely.

A substantial disadvantage of the BSD is, however, the eight-fold computational complexity increase over a standard MSE search.

Improvements in processor technology will, however make the BSD, and more complex psycho-acoustic models, a practical alternative. Currently, the BSD is a primitive model of the human auditory processes - at best it approximates some 15,000 auditory tuning curves with just 64 Critical Band filter functions. The results are however, encouraging and suggest that further work would be worthwhile. In particular, the inclusion of phase in the model should reduce distortion in coded speech and techniques for reducing the complexity of the Critical Band filtering are, clearly, desirable.

7.3 Reductions in Coder Bit-rate

The Prototype waveform (PW) coder, discussed in Chapter 6, aims to exploit the 'pitch periodicity' of speech more explicitly than in CELP coding. By sectioning the input speech into voiced and unvoiced frames, the PW/CELP scheme uses a suitable coding scheme for each frame type and can thus code speech at lower rates than standard CELP. This avoids the adaptive codebook compromise of CELP coders.

For voiced frame coding, the key operation of PW/CELP is the extraction of a suitable prototype to represent the current input speech. The new MSE process, presented, is performed on interpolated speech to give a wider range of prototype shapes, but selects an integer length prototype sampled at 8kHz. While this scheme was shown to be capable of reproducing the required pitch periodicity, it could be improved by using the non-integer pitch schemes described for CELP (see section 3.3). This would, however, substantially increase computation and storage requirements by requiring all operations to be performed at the interpolated sampling rate of 80kHz.

The extracted prototypes are quantised by one of two techniques; a new DFT coefficient quantiser and an Impulsive codebook scheme. The DFT coefficient scheme is simple, but generates a 'hollow' quality in the coded speech. An improved DFT domain quantiser might be based on a mixed codebook structure similar to the Impulsive quantiser. The latter currently produces speech of a higher quality at a lower bit rate.

For interpolation of prototypes across the speech frame, the DFT domain was used as a convenient tool for the manipulation of prototypes of differing pitch periods. The new coefficient interpolation scheme was shown to produce a smooth evolution of prototypes, but, for certain speech sections, the technique tends to lose the attack of the original speech. This problem might be solved by the inclusion of frame 'shape' bits, but this would lead to an inevitable increase in overall bit rate.

In combination with CELP coding for unvoiced frames, the PW/CELP coder was shown to produce good quality coded speech at sub 3kbit/s. The current mixed scheme, however, mixes both open-loop and closed-loop coding schemes. In the long term, and with a substantial increase in complexity, the prototype waveform coder could be made closed loop. Such a scheme would search all quantised prototypes and interpolation schemes for the optimum prototype representation.

A final area of substantial further work would be the real-time implementation of the PW/CELP coder. The current coder is of similar complexity to CELP, but a fully closed-loop PW/CELP coder would create substantial implementation challenges.

7.4 Summary

In summary, this thesis has considered the use of hybrid time-frequency domain techniques to both improve coded speech quality, and reduced

overall bit rate. A novel overlapped codebook technique for frequency domain searched CELP was described. The incorporation of this technique makes Frequency Domain CELP a practical alternative to Time Domain schemes and has the significant advantage of allowing perceptual coding by multiplicative weighting. This has led to the full integration of a perceptual measure, the BSD, into both Time Domain and Frequency Domain architectures. BSD searched CELP has been shown to produce superior speech quality compared to the standard perceptually weighted Time Domain CELP. A new, increased resolution BSD was shown to produce further quality improvements.

In the hybrid PW/CELP, frequency domain techniques were used as a convenient tool for prototype alignment, interpolation and quantisation. The new prototype extraction, interpolation and quantisation schemes allowed the design of a sub-3.2kbit/s coder.

From this work, it is clear that hybrid frequency-time domain coders offer significant advantages over those operating in either domain alone. Future coders will exploit such techniques further, and, in particular, incorporate more sophisticated frequency domain auditory models in conjunction with the standard, time domain LPC techniques.

Chapter 8: Acknowledgments

The author would like to acknowledge the assistance of the following in the work described in this thesis:
Dr R. J. Holbeche, Mr J. D. Martin, Dr J. E. Marshall, Dr R. F. Ormondroyd, Dr C. F. Bore.
I would also like to thank Mr A. G. Grout for his painstaking proof reading of this thesis.

Finally, on a personal note, I should like to thank my family for their continuing support.

This work was supported by Vodafone plc. and S.E.R.C on studentship 89804921.

Appendix I: Publications Arising from this work

1. The Application of the DFT to CELP architectures: IEEE Workshop on Speech Coding for Telecommunications, Whistler, B.C., Canada, September 1991.

2. A Mixed Prototype Waveform / CELP Coder for sub 3kb/s: International Conference on Acoustics, Speech and Signal Processing, April 1993.

The application of the DFT to CELP architectures

I.S.Burnett and R.J.Holbeche
School of Electronic and Electrical Engineering
University of Bath, U.K. *

1 Summary

This paper shows that overlapping codebooks can be used to significantly reduce storage requirements for discrete frequency domain searched Code Excited Linear Prediction (CELP) algorithms. The performance of the analysis-by-synthesis architecture using innovation sequences consisting of few DFT coefficients is also investigated.

2 Overlapping DFT domain Codebooks

In time domain CELP, efficient codebook techniques reduce both codebook search times and storage requirements [1]. A typical overlapped time domain codebook requires only 2088 samples (≈8Kbytes).

A variant on the CELP architecture chooses the optimum innovation sequence in the DFT domain (Figure 1). Since it is impractical to perform the adaptive codebook search in the frequency domain, an efficient closed loop time domain search is retained [1]. Searching the fixed codebook in the frequency domain offers significant computational advantages, however the storage requirements for a frequency domain codebook can be high (e.g.≈320Kbytes for a codebook of 1024 40 sample complex floating point vectors). This section shows that codebook size reductions can be achieved using similar techniques to those used in time domain coders.

In order to transform the codebook search to the frequency domain it is necessary to truncate the IIR inverse short term filter impulse response; in practice, this decays rapidly and can be truncated at 5ms [2]. It is further neccesary to avoid circular convolution by zero-padding the 40 sample impulse response and fixed code to 80 samples. This procedure, however, results in the 40 point DFTs of the time domain sequences being interpolated to 80 points.

To derive an overlapped DFT codebook, the interpolation process must be approximated. Two techniques were used; the first (LONG) generates the codebook as a series of DFTs of long zero-padded gaussian sequences, while the second (CONV) convolves the DFT of a 40 sample step function with a zero-interpolated complex gaussian sequence. In both cases the required 80 point DFTs of real time sequences are generated by employing conjugate symmetry to expand the 40 complex samples from the codebook. This procedure results in a frequency domain codebook requiring only twice the storage of an equivalent length overlapping time domain codebook (2088 complex samples / 16Kbytes), while the DFT domain approximation introduces minimal distortion in the equivalent time sequences.

The overlapped discrete frequency domain coder was simulated using 8KHz sampled speech, 10th order LPC and 160 sample frames divided into four subframes. Table 1 shows the performance of the coder for both overlapped codebooks using an input speech record of 20 male/female sentences from the Harvard list. For a 1024 word overlapped codebook there is no significant degradation in the output speech compared to full frequency domain codebook techniques. There is also no significant performance degradation for small shifts when using small codebooks; this contrasts with results for overlapping time domain codebooks [2].

3 DFT analysis of innovation sequences

The frequency domain treatment of the CELP architecture allows the derivation (using deconvolution by division in the frequency domain) of a pseudo-ideal innovation sequence. As noted in [2] the derived DFT coefficients are not ideal, but, in practice, the resulting time domain sequence gives very high quality speech. The analysis of the pseudo-ideal sequence reveals a number of interesting characteristics.

* This work was supported by Racal Vodafone and U.K. SERC.

The most important DFT coefficients in the innovation sequence spectrum will be those at the peaks of the input weighted speech. Analysis of the use of a limited number of these 'essential' peak coefficients shows that only five need be used to better the performance of time domain CELP. This correlates with similar results in [2] which used coefficients in the SVD domain. Results for various numbers of peak coefficients are shown in the graph of Figure 2.

There is also significant correlation between the peaks chosen from sub-frame to sub-frame. A scheme using one set of peak positions, but retaining the correct values at those positions in each sub-frame, also produces higher performance than CELP algorithms.

In considering these results it should be noted that unquantised coefficient values were used and a coding scheme based on these techniques would probably require a bit rate similar to Multi-Pulse LPC.

References

[1] W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum. Fast methods for the CELP speech coding algorithm,. *IEEE Trans. Acoust., Speech, Signal Processing*, 38(8):1330-1342, August 1990.

[2] I. M. Trancoso and B. S. Atal. Efficient search procedures for selecting the optimum innovation in stochastic coders,. *IEEE Trans. Acoust., Speech, Signal Processing*, 38(3):385-396, March 1990.

Codebook Size	Codebook LONG OVERLAP			Codebook CONV OVERLAP			Time CELP	Freq. CELP
	2	4	8	2	4	8		
128	10.37 10.93	10.38 11.00	10.36 10.90	10.31 10.83	10.31 10.88	10.28 10.79	10.55 10.63	10.44 10.98
1024	11.34 12.15	11.35 12.06	11.31 12.03	11.38 12.15	11.31 12.01	11.32 12.08	11.78 12.06	11.47 12.08
	All Results :- SEG.SNR in (dB) AV.SNR							

Table 1: Comparison of SEGSNRs/SNRs (dB) for overlapping DFT, Time and Full Frequency Domain codebooks.

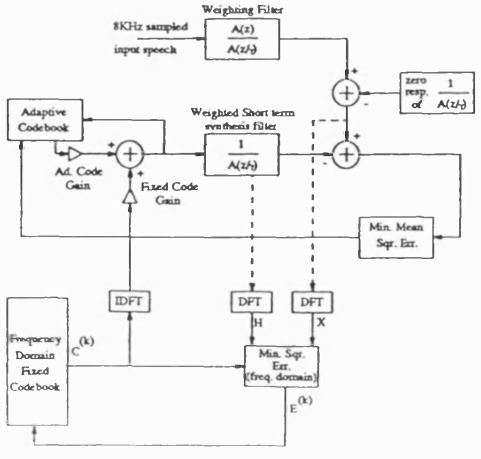


Figure 1: CELP architecture using frequency domain fixed codebook search.

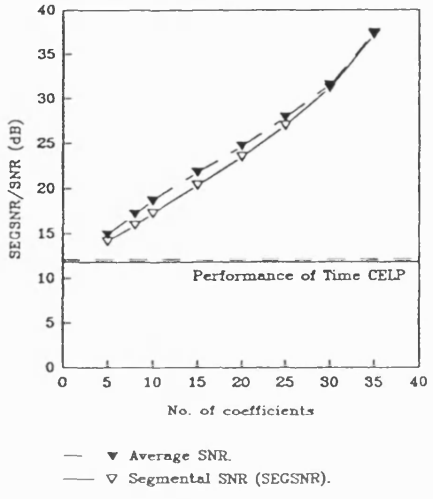


Figure 2: Segmental and Average SNRs vs the number of DFT coefficients in the innovation sequence.

A Mixed Prototype Waveform / CELP Coder for sub 3kb/s

I. S. Burnett and R. J. Holbeche*
School of Electronic and Electrical Engineering
University of Bath, U.K. BA2 7AY

Abstract

CELP Analysis-by-Synthesis speech coders do not make a distinction between voiced and unvoiced speech frames. For sub-3kb/s coding it is necessary to separate unvoiced and voiced frames and code voiced speech using an inherently periodic scheme. This paper addresses these problems by using a prototype waveform coder for voiced frames, while retaining a CELP algorithm for unvoiced frames.

For voiced speech a single 'residual prototype' is selected to represent a section of 25ms. Prototypes are interpolated across the frame to provide a smooth evolution of amplitude and harmonic content. Two coding schemes for the prototypes are discussed; a pitch harmonic scheme operating in the DFT domain, and an impulsive codebook time domain technique. Unvoiced frames are coded using a standard CELP architecture excluding the adaptive codebook search. The overall bit rate using either of the voiced frame coding algorithms is shown to be sub 3kb/s for good communications quality speech.

1. Introduction

Current CELP speech coders operating at 4.8kb/s produce the periodicity necessary for voiced speech using a combination of the fixed codebook and a LTP in the form of an adaptive codebook. At lower rates such an approach produces unacceptably harsh speech as the fixed codebook size is reduced and the code gains are more harshly quantised. So as to generate voiced speech with the required periodicity at low bit rates inherently periodic excitation forms have been suggested [1][2].

In this work, we describe a prototype approach for coding voiced frames, in combination with a simplified CELP coder for unvoiced frames. The combined Prototype Waveform/CELP coder (PW/CELP) interpolates aligned residual prototypes to form the LPC excitation for voiced speech. Unvoiced sections are synthesised by exciting the LPC inverse filter with a gaussian codebook vector.

2. Voiced Frame Coding

The Prototype Waveform coder used for coding voiced speech sections is shown in Figure 1. The following sections describe the fundamental operations performed in this coding technique.

* This work was supported by Vodafone Ltd. and UK SERC.
I.S. Burnett is currently with Bores Signal Processing, Woking, U.K.

2.1 Pitch Determination.

The derivation of pitch synchronous residual prototypes from the input speech requires a reliable method of pitch determination. Pitch determination was an important part of the early speech coders and the technique described here is a progression of the technique developed by Dubnowski et. al. [3] in hardware. A single voiced/unvoiced decision is also made for each frame, based on the normalised value of the autocorrelation for the chosen pitch. The threshold for a voiced decision (26% of R(0)) was made lower than in [3]. This bias was considered preferable, since a voiced frame coded using a gaussian excitation model is likely to produce more unpleasant auditory distortion than an unvoiced frame with added periodicity.

2.2 Prototype Derivation

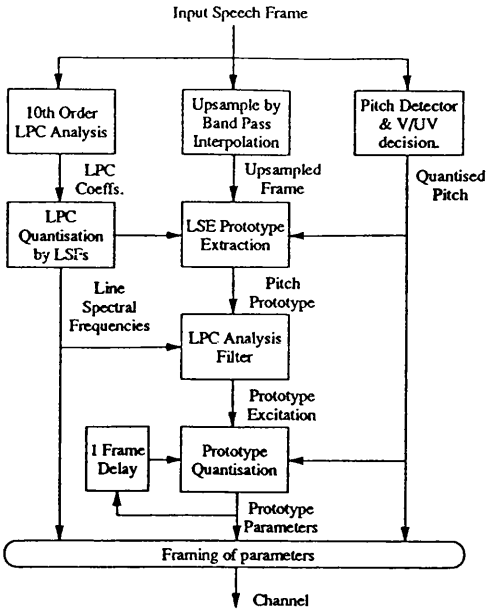
Extraction of a prototype from a given voiced frame is performed using a Least Squares Error calculation between a concatenated repeated prototype and the input speech frame. For the purposes of prototype extraction the 8kHz sampled input speech is upsampled by bandpass interpolation to 80kHz. Protoypes are then selected from the upsampled speech frame and concatenated to form the candidate prototype frame. The upsampling is thus used to increase the number of candidate prototypes and hence increase the likelihood of finding a prototype which concatenates smoothly.

Prototypes are derived from the interpolated frame by extracting τ samples from a point *start* in the interpolated frame. The prototype is sampled at the 8kHz rate such that the prototypes $p(n)$ are defined as:

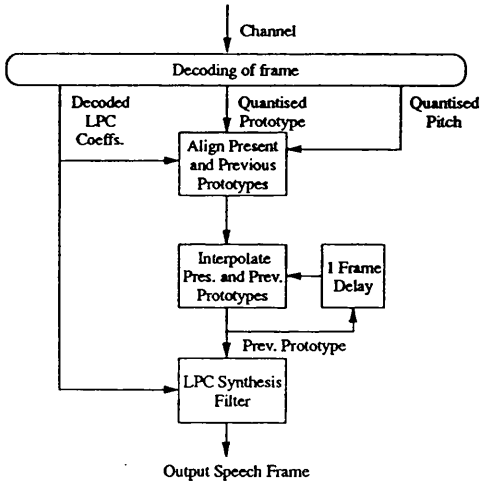
$$p(n) = s_i(start + n * M) \quad n = 1,.....,\tau \qquad (1)$$

where M is the interpolation factor and, in this case, $M=10$.

This 'pitch period prototype' is then repeated to a frame length L_f (in this case 200 samples) to produce an 'extended prototype frame'. The prototypes within the prototype frame are arranged to be synchronous with the base prototype in terms of its frame position. The mean square error, E_p, between the input speech frame, S, and the extended prototype frame is then calculated. This calculation is repeated for all possible prototype starting points and the prototype minimising the value of E_p is chosen as the prototype to represent the current voiced frame. It was found that the inclusion of a gain term,



(a) Prototype Waveform Transmitter Structure.



(b) Prototype Waveform Receiver Structure

Figure 1: The Prototype waveform encoding technique for voiced speech frames.

G_p, in the prototype extraction process enhanced synthesised speech quality. This caters for frames having a wide variation in prototype amplitude and is computed using the standard least squares gain optimisation.

The final prototype derived from the extraction process is thus:

$$p_m(n) = G_p * p(n), \quad n = 1,2,.....\tau \tag{2}$$

An example of a set of gain adjusted prototypes, extracted from successive voiced speech frames, are shown in Figure 2(b). Future operations on the prototypes are performed on the DFT of $p_m(n)$ which is denoted $\mathbf{P}_m(k)$.

2.3 Derivation of the 'Residual Prototype'

The final operation of the prototype extraction is to derive an LPC residual of the prototype. The residual is produced by filtering the prototype with the standard LPC filter (using the coefficients $a(k)$, $k=1,..,p$ calculated for the current frame). It is, however, necessary to ensure that the residual reproduces the continuous nature of the extended prototype frame. The residual prototype is thus calculated from a periodically extended prototype section of length $\tau+p$. This operation results in a residual prototype devoid of the formant structure determined by the LPC analysis of the current frame.

2.4 Alignment of Residual Prototypes.

Since residual prototypes of successive frames will not generally be 'in phase' it is necessary to time align the present prototype prior to interpolation. Alignment of the prototypes allows smooth interpolation to be performed and removes the necessity to transmit 'position information' for prototypes. For the purposes of the time alignment the 'previous' residual prototype is regarded as being the final τ_{m-1} samples of the previously interpolated excitation frame.

The alignment operation is performed in a similar manner to that suggested by Kleijn[1]. Both the current and previous residual prototypes are spectrally weighted using the current frame's LP coefficients. The weighted prototype DFTs ($\mathbf{Q}_m(k)$ and $\mathbf{Q}_{m-1}(k)$) are then aligned by finding the normalised time shift, θ, which maximises the cross correlation:

$$\theta = \underset{\theta'}{argmax} \sum_{k=0}^{\tau} \mathrm{Re}\left[\mathbf{Q}_m(k)\mathbf{Q}^*_{m-1}(k)\, e^{j\,2\pi k\,\theta'} \right] \quad for\ \theta' = 0,...,1 \tag{3}$$

The coefficients of the aligned present residual prototype are then computed as:

$$\mathbf{P}'_m(k) = \mathbf{P}_m(k)\, e^{j\,2\pi k\theta} \quad for\ k = 0,1,.....\tau \tag{4}$$

2.1 Interpolation of Residual Prototypes

Residual prototype interpolation is performed on a pitch period basis in the DFT domain. Interpolation of the transform coefficients of the present and previous aligned residual prototypes, $\mathbf{P}_m(k)$ is equivalent to linear interpolation of the time domain prototypes. The interpolation process is thus an evolution of both the prototype length and amplitude characteristics. Since the pitch period may alter over the interpolation interval the number of DFT coefficients describing the prototypes will alter over the interpolation

interval. It is thus convenient to define a 'prototype length' counter such that:

$$C_p = \sum_{i=0}^{p-1} \tau_i \quad for\ p = 0,1,\ldots\ and\ where\ \tau_0 = \tau_{m-1} \tag{5}$$

For an interpolation interval, L_i, a linear interpolation coefficient, $0 \le \alpha \le 1$, is then defined as:

$$\alpha = \frac{C_p + \tau_p}{L_i} \tag{6}$$

The interpolation coefficient describes the 'contribution' of the present prototype such that the pitch period will evolve as:

$$\tau_{p+1} = (1-\alpha)\tau_{m-1} + \alpha\tau_m \tag{7}$$

and the time domain reconstructed excitation is then described by:

$$e_i(t + C_p) = \mathrm{Re} \sum_{k=1}^{\tau_p} \left((1-\alpha)P'_{m-1}(k) + \alpha P'_m(k)\right) e^{j\frac{2\pi kt}{\tau_p}}$$
$$for\ t = 0,1,\ldots\ldots,\tau_p \tag{8}$$

The sequence of operations defined in equations (5-8) are then repeated until C_p exceeds the interpolation frame length. Throughout the process the conjugate symmetry of the DFT of a real sequence is maintained even though the number of points of each effective IDFT varies.

In practice, it was found that interpolation across the whole excitation frame resulted in a 'slurring' of some reproduced speech. This is not unexpected since the prototypes can be derived from any section of the input speech frame and the interpolation process is thus limited to 75% of the excitation frame.

3. Quantisation of Prototypes

Two quantisation techniques have been implemented; one operates in the frequency domain and the other in the time domain. The major challenge presented by the quantisation is the variable length of the prototypes (from 16 to 147 samples).

3.1 Frequency Domain Quantiser

In the DFT domain the residual prototype is represented by τ complex coefficients representing the contributions of the Fourier basis functions. The Frequency domain quantisation technique divides the spectrum of the prototype into two regions separated by 1kHz. Frequency components below 1kHz are coded as four complex coefficients derived using the standard DFT basis functions. The spectral region beyond 1kHz is represented using two 'enhanced bandwidth' coefficients similar to those described by Gupta and Atal [4]. The 'enhanced bandwidth' coefficients are derived using reduced width windowing of the original basis functions.

The six coefficient pairs are then quantised differentially, with respect to the previous prototype's coefficient values, using six bits per complex coefficient. For the purposes of differential coding the prototype is aligned with the previous prototype using the technique described in section 2.4.

3.2 Impulsive Quantiser

The impulsive quantiser operates in the time domain and is similar to the quantisation technique described by Granzow et. al. [2]. The quantisation consists of two searching of two codebooks populated by impulses and the calculation of a differential gain component from the previously quantised prototype.

The quantisation procedure can be summarised by:

$$\mathbf{u}_m(n) = \alpha_0 \mathbf{v}_0(n) + \alpha_1 \mathbf{v}_1(n) + \beta\, \mathbf{u}_{m-1}(n)$$
$$for\ n = 0,1,\ldots\ldots\tau_m \tag{9}$$

where $\mathbf{v}_0(n)$ and $\mathbf{v}_1(n)$ two codebook vectors and $\mathbf{u}_m(n)$ and $\mathbf{u}_{m-1}(n)$ are the new and previously quantised prototype, respectively. The gain terms α_0 , α_1 and β are similar to the gain terms calculated in the CELP search. The first codebook vector $\mathbf{v}_0(n)$ is derived from a codebook of 128 vectors consisting of single delta impulses of unit amplitude. The second codebook vector $\mathbf{v}_1(n)$ is derived from a 128 vector ternary codebook formed by centre clipping a gaussian codebook.

So as to derive a useful differential contribution the present, unquantised and previously quantised prototypes are aligned using the time alignment operation discussed in section 2.4. The gain terms β, α_0 , α_1 are then computed by the standard MSE gain calculation and quantised using 5 bits per parameter.

4. Unvoiced Frame Coding

Unvoiced frames are coded using a standard CELP search of a 128 vector overlapped gaussian codebook. Each 200 sample frame is divided into three sub-frames and the gains for each sub-frame are coded with 5 bits. The total bit rate for coding of an unvoiced frame excitation is thus 36 bits and is comparable with the total bits required for the coding of voiced frames using the impulsive codebook scheme. The codebook searches are performed using the standard perceptually weighted squared error measure.

5. Combination of Prototype and CELP algorithms

The output excitation vectors from both the voiced and unvoiced frame coders are inverse filtered by the common LPC inverse filter. So as to ensure a smooth transition between unvoiced and voiced sections, the voiced frame's 'previous excitation prototype' is taken to be the last τ_{m-1} samples of the previous unvoiced frames excitation.

The combined algorithms are forthwith referred to as a mixed prototype waveform / CELP scheme (PW/CELP). In terms of coding, the two constituent coders operate independently for voiced and unvoiced frames, however, there is a common requirement for the derivation and quantisation of the 10 LP coefficients. These are coded using Line Spectral Frequencies. which can be coded efficiently using split-VQ techniques in 25 bits as described in [5].

6. Results of PW/CELP

The bit allocation for voiced and unvoiced schemes using the various parameter quantisation schemes is described in Table 1.

Coded speech from the PW/CELP algorithm is shown in Figure 2. The waveforms show that the coder generally reproduces voiced speech well, with the basic pitch periods evolving similarly to the input speech. In transitional unvoiced/voiced sections, however, it was found that the interpolation procedures slow the coders response and the synthesised speech fails to reproduce the fast waveform changes of the input. These phenomena degrade the synthesised speech quality but appear unavoidable when a long frame length is used to reduce overall bit rate.

Since objective measures are not applicable to the non-synchronous PW/CELP coder informal listening tests have been performed. These show that PW/CELP is currently between low rate vocoders and the 4.8kb/s CELP algorithms in performance. The Impulsive Codebook, time domain prototype quantiser currently produces better quality speech at a lower bit rate than the frequency domain scheme. The former produces harsher speech but this is preferable to the tonal distortions generated by the frequency domain quantisation. Both techniques, however, retain a naturalness in the speech absent from low rate vocoders.

7. Conclusions

This work has shown that prototype coding techniques, when integrated with standard CELP, can offer good quality coding of speech at rates considerably lower than those of CELP alone. We are currently investigating new prototype interpolation techniques which will improve the coders response to unvoiced/voiced transitions. Also, the prototype coder is currently open-loop and we are considering a closed-loop architecture which, at the expense of complexity, should produce further improve speech quality.

References

[1] W. B. Kleijn, "Continuous Representations in Linear Predictive Coding," *Proc. Int. Conf. on Acoust., Speech and Signal Proc.*, pp. 201-204, May 1991.

[2] W. Granzow, B. S. Atal, K. K. Paliwal and J. Schroeter, "Speech Coding at 4KB/S and Lower Using Single-Pulse and Stochastic Models of LPC Excitation," *Proc. Int. Conf. Acoust., Speech and Signal Proc.*, pp. 217-220, May 1991.

[3] J. J. Dubnowski, R. W. Schafer and L. R. Rabiner, "Real-Time Digital Hardware Pitch Detector," *IEEE Trans. on Acoust., Speech and Signal Proc.*, Vol. 24, No. 1, pp. 2-9, Feb. 1976.

[4] S. K. Gupta and B. S. Atal, "Efficient Frequency-Domain Representation of LPC-Excitation," *IEEE Workshop on Speech Coding for Telecomms.: Digital Voice for the Nineties*, pp. 64-65, Sept. 1991.

[5] K. K. Paliwal and B. S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame," *Proc. Int. Conf. Acoust., Speech and Signal Proc.*, pp. 661-664, May 1991.

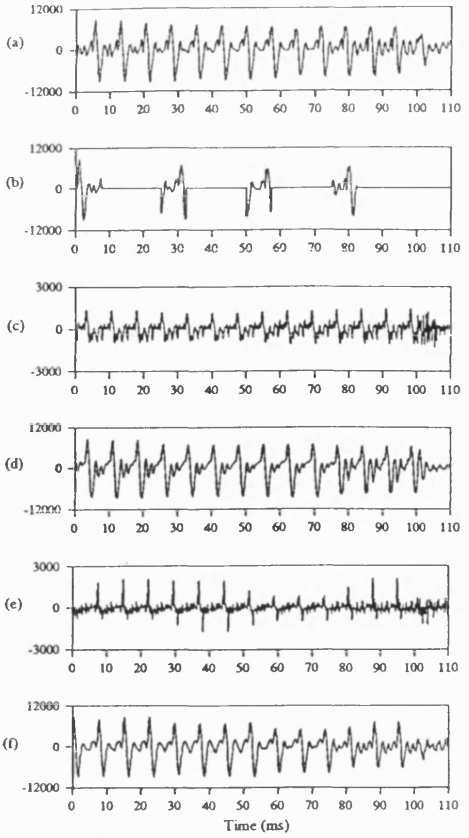


Figure 2: Waveforms from a 'voiced' speech section coded using the PW/CELP coder: (a) Input speech, (b) Extracted prototypes, (c)&(d) Quantised residual and output speech generated by the Frequency Domain quantiser, and (e)&(f) by the impulsive codebook quantiser. For all results all LP coefficients. and gains fully quantised.

	Impulsive Codebook	Frequency Domain
LSFs (Split-VQ / Fed Std 1016)	25/34	25/34
Voiced Frame: PW Coder:		
Pitch	7	7
Voiced/Unvoiced Decision	1	1
Prototype Codebook Indexes (2)	14	-
Gain Terms	15	36
Voiced Frame Total:	62/71	69/78
Unvoiced Frame: CELP:		
Codebook Indexes(3)	21	21
Codebook Vector Gains(3)	15	15
Unvoiced Frame Total:	61/70	61/70
Max. Bits/Frame:	62/71	69/78
Overall Bit Rate (Bits/sec) :	2480/2840	2760/3120

Table 1: Bit Allocations for PW/CELP.